
Getting Started : W-8037/8337/8737

The Web PAC

By ICP DAS CO. , LTD., Apr. 2005, All Rights Reserved

ICP DAS CO., LTD. would like to congratulate you own your purchase of our Web Programmable Automation Controller (Web PAC) – Wincon-8037/8337/8737.

Legal Liability

ICP DAS CO., LTD. assumes no liability for any and all damages that may be incurred by the user as a consequence of this product. ICP DAS CO., LTD. reserves the right to change this manual at any time without notice.

ICP DAS CO., LTD. constantly strives to provide our customers with the most reliable and accurate information possible regarding our products. However, ICP DAS CO., LTD. assumes no responsibility for its use, or for any infringements of patents or other rights of third parties resulting from its use.

Trademark & Copyright Notice

The names of products and name of company are used for identification purposes only, and are the registered trademarks of their respective owners or companies.

Development Software

Two options:

- ISaGRAF: Ver. 3.4x , IEC61131-3 standard. Languages: LD, ST, FBD, SFC, IL & FC or
- Non-ISaGRAF: Microsoft EVC++4.0 or VS.NET 2003 (VB.NET , C#.net)

Reference Guide

- ISaGRAF English User's Manual:
CD-ROM: \napdos\isagraf\wincon\english_manu\ "user_manual_i_8xx7.pdf"
- ISaGRAF中文使用手冊:
CD-ROM: \napdos\isagraf\wincon\chinese_manu\ "chinese_user_manual_i_8xx7.pdf"
- CD-ROM on the Internet:
Newly updated ISaGRAF IO libraries, drivers and manuals can be found at
<http://www.icpdas.com/products/PAC/i-8000/isagraf.htm>

Technical Service:

Please contact local agent or email problem-report to service@icpdas.com .

FAQ on the Web - <http://www.icpdas.com/faq/isagraf.htm>

New information can be found at www.icpdas.com

Copyright Apr. 2005, by ICP DAS CO., LTD. All Rights Reserved.

Table of Contents

GETTING STARTED : W-8037/8337/8737	1
THE WEB PAC.....	1
<i>Legal Liability.....</i>	<i>1</i>
<i>Trademark & Copyright Notice.....</i>	<i>1</i>
<i>Development Software</i>	<i>1</i>
<i>Reference Guide.....</i>	<i>1</i>
<i>Technical Service:.....</i>	<i>1</i>
TABLE OF CONTENTS	2
REFERENCE GUIDE.....	5
SPECIFICATIONS: WINCON-8037 / 8337 / 8737	6
CHAPTER 1: TYPICAL APPLICATION.....	8
1.1: WINCON WEB HMI ON THE LOCAL ETHERNET	8
1.2: WINCON WEB HMI ON THE INTERNET	9
1.3: DOWNLOAD DATA FILE FROM THE WINCON CONTROLLER.....	10
1.4: PLC WITH LARGE STORAGE MEMORY.....	10
1.5: MULTI-HMI & REMOTE I/O.....	11
1.6: SMS	11
1.7: DATA EXCHANGE THROUGH ETHERNET & RS485.....	12
1.8: MOTION CONTROL.....	12
1.9: MODBUS CONVERTER OF I-7000 & I-87K I/O.....	13
1.10: MODBUS MASTER.....	14
1.11: DOWNLOAD & MONITORING VIA MODEM_LINK.....	14
CHAPTER 2: SOFTWARE INSTALLATION	15
2.1: STEP 1 - INSTALLING THE ISAGRAF SOFTWARE	15
2.1.1: <i>When closing my ISaGRAF window on windows 2000, it holds. Why ?</i>	<i>17</i>
2.1.2: <i>One Quick way to avoid the “hold” problem on windows 2000.....</i>	<i>18</i>
2.2: STEP 2 - INSTALLING THE ICP DAS UTILITIES FOR ISAGRAF	19
2.3: STEP 3 - INSTALLING THE WEB PAGE EDITOR	20
CHAPTER 3: SETTING UP A WEB HMI DEMO.....	21
3.1: WEB DEMO LIST	21
3.2: STEPS TO SET UP A WEB HMI DEMO	22
3.2.1: <i>Step 1 - Setup The Hardware.....</i>	<i>22</i>
3.2.2: <i>Step 2 - Make Sure The Right ISaGRAF Driver Installed.....</i>	<i>22</i>
3.2.3: <i>Step 3 - Setting The Web Options</i>	<i>23</i>
3.2.4: <i>Step 4 - Download ISaGRAF Project.....</i>	<i>23</i>
3.2.5: <i>Step 5 - Download Web Pages To The Wincon.....</i>	<i>27</i>
3.2.6: <i>Step 6 - Show Time</i>	<i>27</i>
CHAPTER 4: PROGRAMMING A WEB HMI EXAMPLE.....	28

4.1: WRITING A SIMPLE ISAGRAF PROGRAM	28
4.1.1: <i>Open ISaGRAF-Project Management</i>	29
4.1.2: <i>Creating An ISaGRAF User's Group</i>	30
4.1.3: <i>Creating A New ISaGRAF Project</i>	30
4.1.4: <i>Declaring The ISaGRAF Project Variables</i>	31
4.1.5: <i>Assign Modbus Network Address No to Variables</i>	35
4.1.6: <i>Create The LD - "LD1" Program</i>	37
4.1.7: <i>Edit The "LD1" Program</i>	38
4.1.8: <i>Connecting The I/O</i>	42
4.2: COMPILING & SIMULATING THE EXAMPLE PROJECT	44
4.3: DOWNLOAD & DEBUG THE EXAMPLE PROJECT	48
4.4: DESIGN THE WEB PAGE	53
4.4.1: <i>Step 1 – Copy The Sample Web HMI pages</i>	53
4.4.2: <i>Step 2 – Building The Main.htm</i>	54
4.4.3: <i>Step 3 – Adding Control Code To The Main.htm</i>	59
4.4.4: <i>Step 4 – Download Web HMI Pages To The Controller</i>	66
CHAPTER 5: WEB HMI BASICS.....	69
5.1: BASIC FILES FOR THE WEB HMI	69
5.2: LOGIN.HTM	70
5.3: MENU.HTM	71
5.4: MAIN.HTM	73
5.4.1: <i>A Simple Main.htm Example</i>	73
5.4.2: <i>More About The refresh_data() Function And Dynamic Data</i>	75
5.4.3: <i>Post Data To The Controller</i>	80
5.5: MULTI-PAGES	84
5.5.1: <i>Level 2 And Level 3 Page</i>	84
5.5.2: <i>Switch One Page To One Another Page</i>	85
5.6: WEB SECURITY	86
CHAPTER 6: VB.NET PROGRAM RUNNING IN WINCON ACCESS TO ISAGRAF VARIABLES	88
6.1 CREATE A NEW SDE PROJECT	88
6.2 ADD PROJECT REFERENCE FOR AN APPLICATION	89
6.3 BUILDING AN APPLICATION PROGRAM	91
6.4 EXECUTE THE APPLICATION IN THE WINCON-8XX7 PLATFORM	92
6.5 EXAMPLES	92
6.6 QUICKERNET.DLL	92
6.6.1 <i>Digital R/W Functions</i>	92
6.6.2 <i>Analog R/W Functions</i>	94
CHAPTER 7: EVC++ PROGRAM RUNNING IN WINCON ACCESS TO ISAGRAF VARIABLES	96
CHAPTER 8: INDUSOFT PROJECT RUNNING IN WINCON ACCESS TO ISAGRAF VARIABLES	98
CHAPTER 9: ISAGRAF EXAMPLE PROGRAM.....	108
9.1: GET ON-LINE HELP	108

9.2: INSTALLING THE ISAGRAF PROGRAMMING EXAMPLES	111
CHAPTER 10: PROGRAMMING W-8X37 BY NON-ISAGRAF	114
10.1: SET WHMI.EXE TO BE FIRST AUTO-EXECUTE PROGRAM	114
10.2: SETTING UP AN EVC++ WEB HMI DEMO	115
10.2.1: <i>Step 1 - Setup The Hardware</i>	115
10.2.2: <i>Step 2 - Make Sure Whmi.exe Installed</i>	115
10.2.3: <i>Step 3 - Setting The Web Options</i>	115
10.2.4: <i>Step 4 - Download And Run EVC++ Demo Program</i>	116
10.2.5: <i>Step 5 - Download Web Pages To The Wincon</i>	117
10.2.6: <i>Step 6 - Show Time</i>	117
10.3: EVC++ FUNCTION TO ACCESS TO THE WEB HMI	118
10.4: VB.NET FUNCTION TO ACCESS TO THE WEB HMI	123
APPENDIX A: HARDWARE SYSTEM & SETTING	124
A.1: APPLYING CORRECT POWER SUPPLY	124
A.2: MODIFY THE NET-ID & MODBUS RTU PORT SETTING	125
A.3: SETTING THE IP ADDRESS FOR THE W-8x37	126
A.4: CONNECTING YOUR PC TO THE W-8x37 ETHERNET PORT	127
A.5: MULTI-CLIENTS CONNECTION TO THE W-8x37 ETHERNET PORT	128
A.6: CONNECTING YOUR PC TO THE W-8x37 COM2 OR COM3	128
A.7: DELETING THE ISAGRAF PROJECT FROM THE W-8x37	129
A.8: LINKING I-7000 AND I-87K MODULES FOR REMOTE I/O	130
A.9: LINKING TO AN HMI INTERFACE DEVICE	131
A.10: LINKING TO OTHER MODBUS DEVICES	132
APPENDIX B: UPDATE WINCON'S ISAGRAF DRIVER TO HIGHER VERSION	133
APPENDIX C: DIMENSION	137
C.1: WINCON-8037	137
C.2: WINCON-8337	138
C.3: WINCON-8737	139

Reference Guide

ISaGRAF English User's Manual:

CD-ROM: \napdos\isagraf\wincon\english_manu\ "user_manual_i_8xx7.pdf"

ISaGRAF 中文使用手冊:

CD-ROM: \napdos\isagraf\wincon\chinese_manu\ "chinese_user_manual_i_8xx7.pdf"

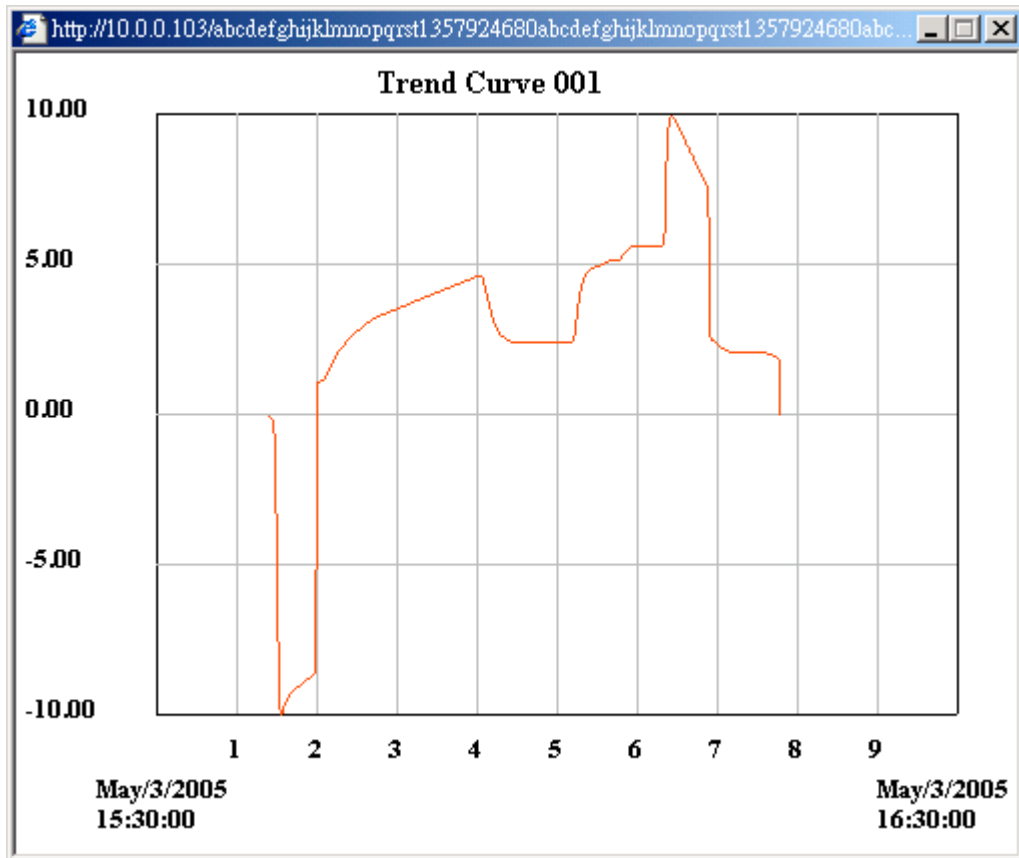
Resource on the Internet:

Newly updated ISaGRAF IO libraries, drivers and manuals can be found at
<http://www.icpdas.com/products/PAC/i-8000/isagraf.htm>

Get sample and display trend curve on the Web HMI:

The Web HMI can display trend curve similar as below figure.

Please refer to CD-ROM: \napdos\isagraf\wincon\english_manu\ "trend_curve.pdf"



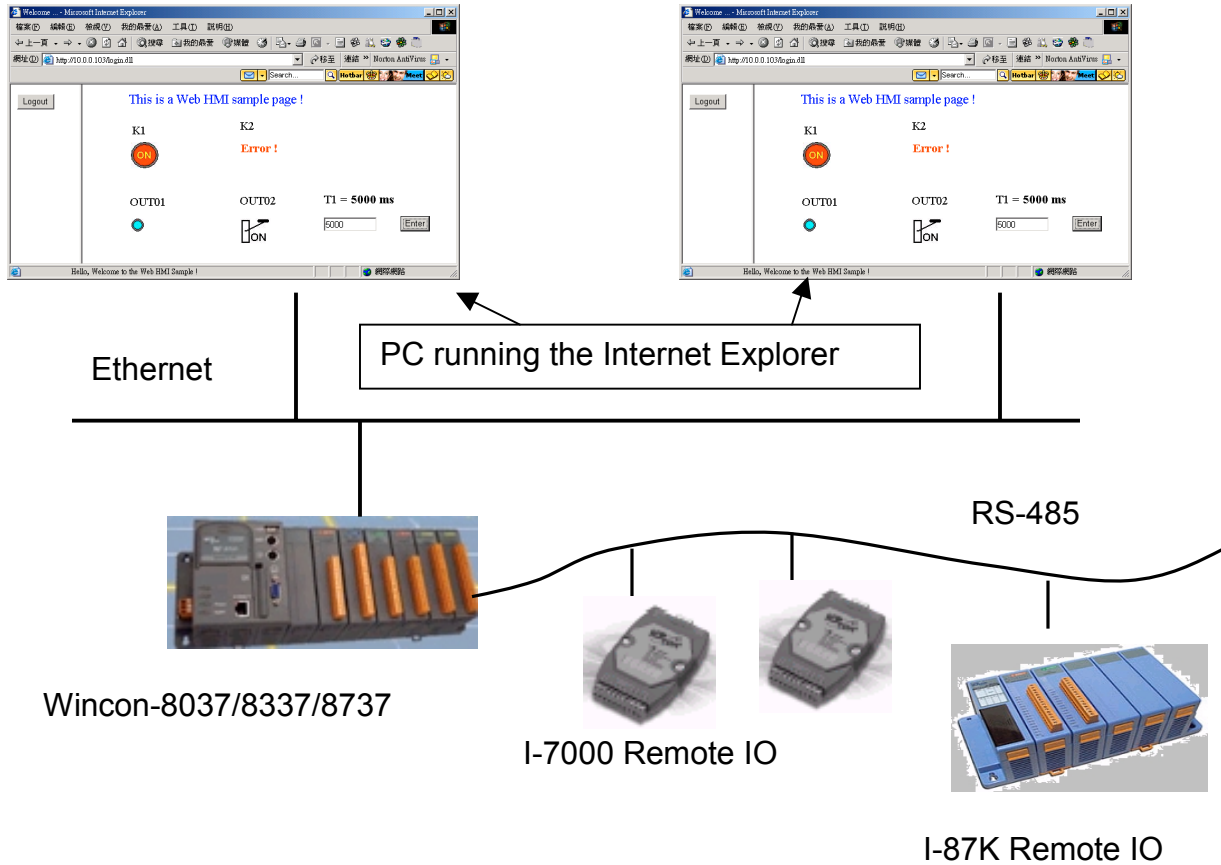
Specifications: Wincon-8037 / 8337 / 8737

Development software	Two options
1. ISaGRAF	IEC61131-3 standard. Languages: LD, ST, FBD, SFC, IL & FC
2. non-ISaGRAF	Microsoft EVC++4.0 or VS.NET 2003 (VB.NET , C#.net)
Web HMI	
Basic	Supports Web HMI function, PC running Internet Explorer can access to the Wincon-8037/8337/8737 via Local Ethernet or Internet, monitoring and Control.
Security	Three Level username and password protection
Dial Modem	PC running Internet Explorer & dial modem to the internet can also access to the Wincon-8037/8337/8737, monitoring and Control
Power supply	
Power requirements	10 to 30VDC (unregulated)
Power consumption	25W (when I/O slots are empty)
Protection	Built-in power protection & network protection circuit
General environment	
Operating temperature	-25°C to +75°C
Storage temperature	-30°C to +85°C
Humidity	5 to 95 % (non-condensed)
System	
CPU	Intel Strong ARM CPU, 206MHz, or compatible
Watchdog timer	Built-in Watchdog Timer
Real time clock	Year-2000 compliance. Gives hour, minute, sec, date of week, date of month, month & year
SDRAM	64Mbytes
FLASH Memory	32Mbytes for OS image
Compact Flash Card	One Compact Flash slot: CF memory card is 128Mbytes or more
EEPROM	16Kbytes
I/O slots	W-8037: No IO slot , W-8337: 3 slots , W-8737: 7 slots Accept I-8K parallel & I-87K serial I/O boards
VGA Port	1 VGA port: 320x240x16 to 1024x768x16, default is 640x480x16
PS/2 Port	2 PS/2 ports: keyboard and mouse
USB port	1 USB 1.1 Host port for USB drive or USB mouse
Reset Button & Led	1 reset button & 1 power Led
Unique Serial Number	64-bit hardware unique serial number
NET ID	From 1 to 255, set by software
Serial ports	
COM1	Internal use for I-87K serial IO boards for W-8337 & W-8737, W-8037 has no COM1
COM2	RS232: full modem signals, Speed: 115200 bps max.
COM3	RS-485, Speed: 115200 bps max. D+, D-
Ethernet	10M bps, NE2000 compatible, 10 BaseT, Program download port.

Motion control	
	The W-8337/8737 can integrate with one I-8091(2-axes) or two I-8091(4-axes) motion board to do motion control.
Counters	
Serial D/I counter	Counter input: 100Hz max. Counter value: 0 to 65535 (16 bit) Optional serial I-87K D/I boards: i-87051, 87052, 87053, 87054, 87055, 87058, 87063
Remote D/I counter	All remote I-7000 & I-87K D/I modules support counters. 100Hz max. Counter value: 0 to 65535 (16 bit)
High speed counter	i-87082: 100kHz max. 32 bit, i-8080: 450kHz max. 32 bit
Protocols	
Modbus serial protocol	COM2 default for connecting ISaGRAF, PC/HMI & MMI panels.
Modbus TCP/IP protocol	Ethernet port for connecting ISaGRAF & PC/HMI
Web HMI protocol	Ethernet port for connecting PC running Internet Explorer
Remote I/O	COM3 supports I-7000 I/O modules & (I-87K base + I-87K serial I/O boards) as remote I/O. Max. 255 I/O modules for one controller
Modbus slave I/O devices	COM2 or COM3 supports Modbus master protocol to connect to other Modbus slave I/O devices
Ebus	A software mechanism built in Ethernet port to exchange data between ICP DAS's ISaGRAF Ethernet controllers.
SMS: Short Message Service	COM2 (or COM5 if found in one extra I-8112 board) can link to a GSM modem to support SMS. User can request data or control the controller by cellular phone. The controller can also send data & alarm to user's cell. phone. Optional GSM modem: M1206: GSM 900/1800 MHz
User defined protocol	User can write his own protocol applied at COM2, COM3, (& COM5 to ... if multi-serial port boards are plugged).
Modem_Link	Supports PC remotely download & monitor W-8037/8337/8737 through a normal modem (link to COM2).
MMICON / LCD	COM2 supports ICP DAS's MMICON. The MMICON is featured with a 240 x 64 dot LCD and a 4 x 4 Keyboard. It can display picture, string, integer, float, and input a character, string, integer and float.
Redundant Bus7000	Two ISaGRAF controllers can link to remote I-7000 & I-87K I/O modules at the same time. Only one controller is active to control these remote I/Os. If one is dead, the other one will take over the control of remote I/Os.
File Access	
	The Compact Flash memory can be used for storing run time data or any controller setting. The size depends on the memory size of the CF card. (default is 128Mbytes)

Chapter 1: Typical Application

1.1: Wincon Web HMI On The Local Ethernet

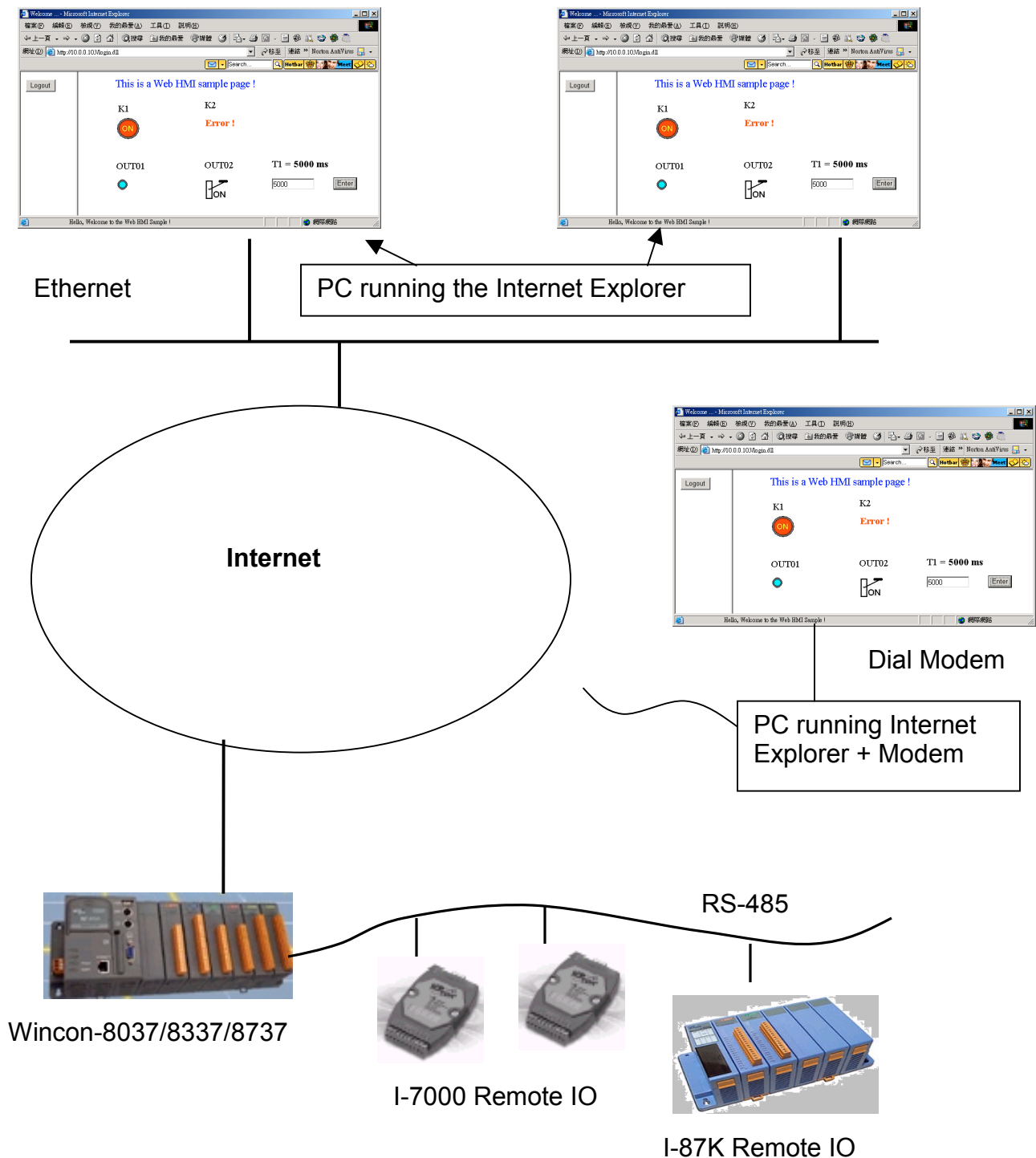


Free HMI.

No extra software to be installed on the PC.

Just Running Microsoft Internet Explorer.

1.2: Wincon Web HMI On The Internet



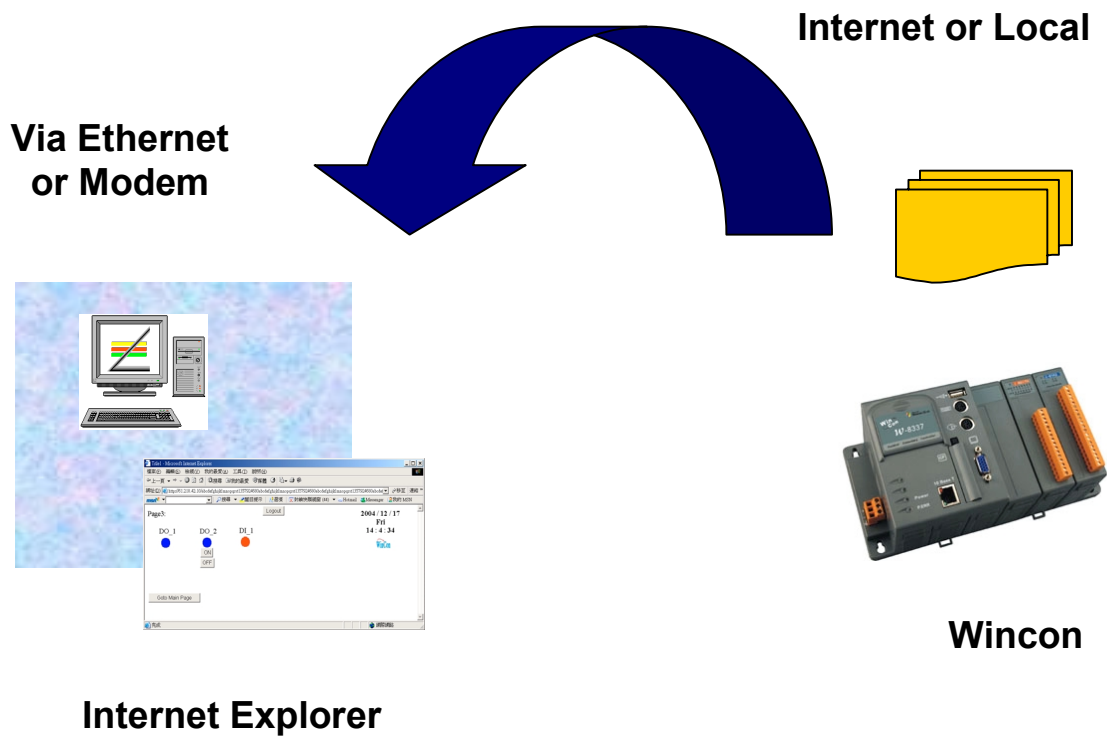
Monitoring & Control Everywhere !

Free HMI.

No extra software to be installed on the PC.

Just Running Microsoft Internet Explorer

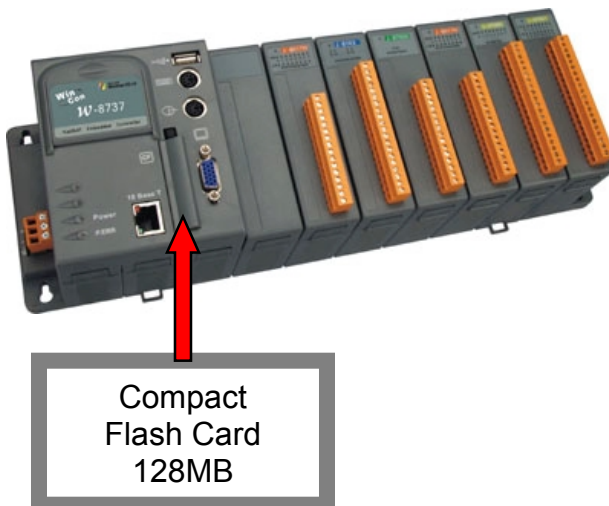
1.3: Download Data File From The Wincon Controller



1.4: PLC With Large Storage Memory

Supporting ISaGRAF IEC61131-3 programming language.
Data can save in the Compact Flash Disk – 128MB or more.

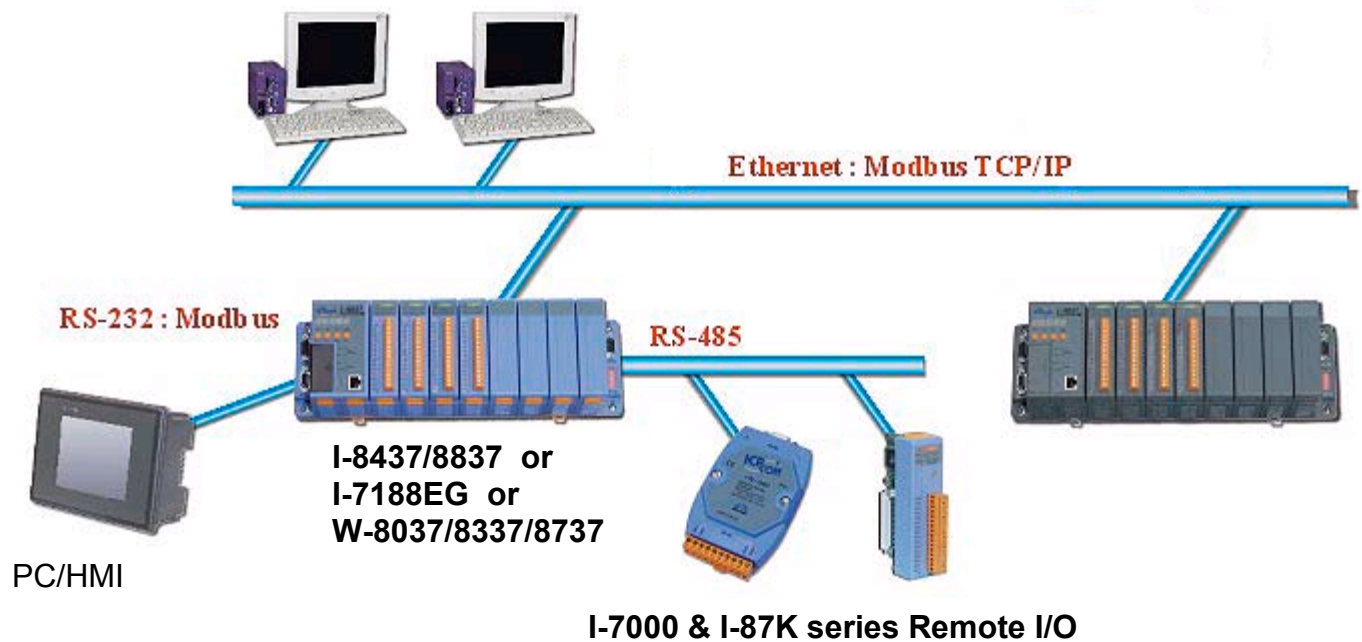
W-8037/8337/8737



IEC61131-3 Standard
Ladder, SFC, ST, FBD, IL & FC

1.5: Multi-HMI & Remote I/O

HMI & SCADA software



1.6: SMS



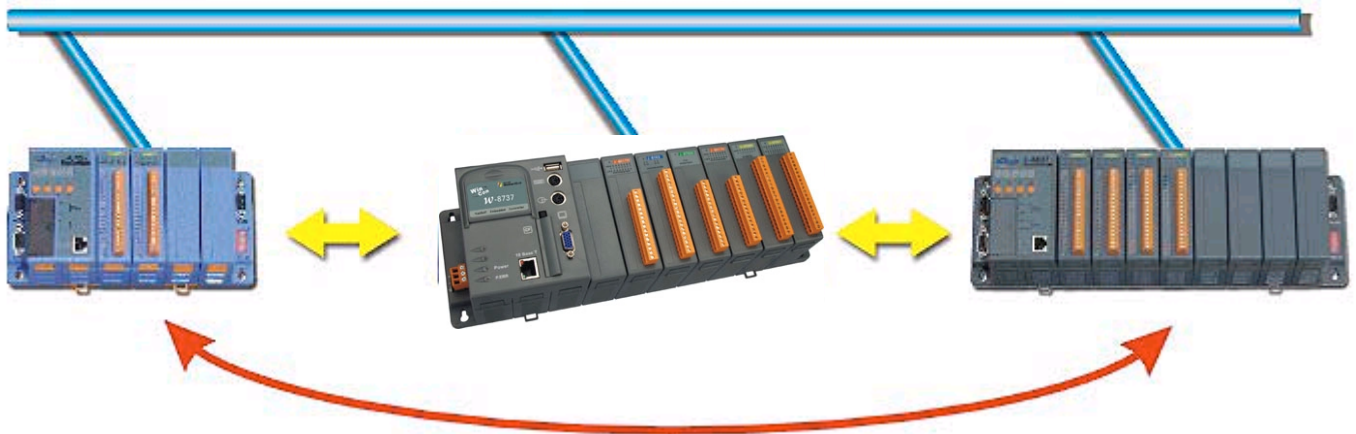
1.7: Data Exchange through Ethernet & RS485

Data Exchange

Controller to controller data exchange

Ethernet: W-8037/8337/8737, I-8437/8837 & I-7188EG

RS485: I-8417/8817/8437/8837, I-7188EG & I-7188XG



1.8: Motion Control

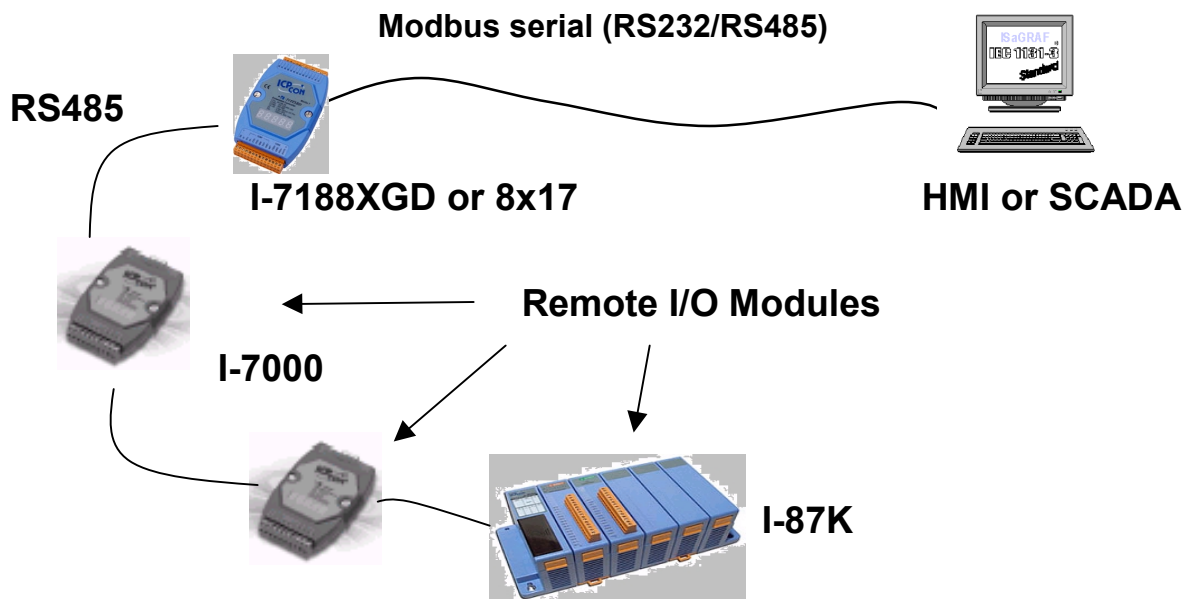
One I-8091 can control 2-axes: X-Y plane, or 2 axes independent

Two I-8091 can control 4-axes: X-Y plane + 2 axes independent, or 4 axes independent

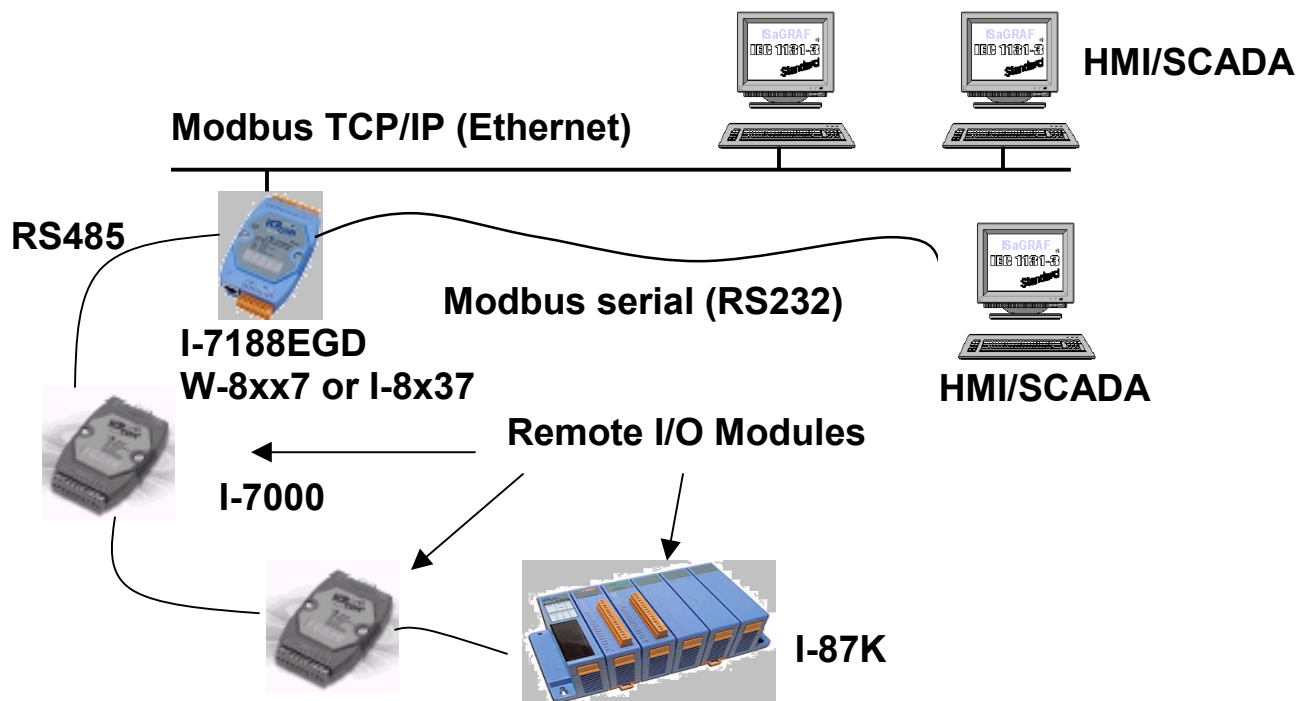


1.9: Modbus Converter Of I-7000 & I-87K I/O

I-7188XG & I-8417/8817 can be a Modbus RTU serial converter of I-7000 & I-87K series I/O modules.

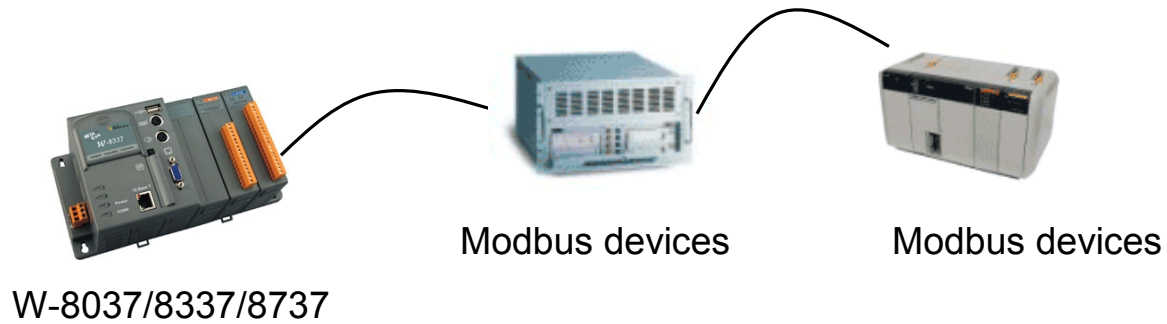


W-8037/8337/8737, I-7188EG & I-8437/8837 can be a Modbus RTU serial & TCP/IP converter of I-7000 & I-87K series I/O modules.

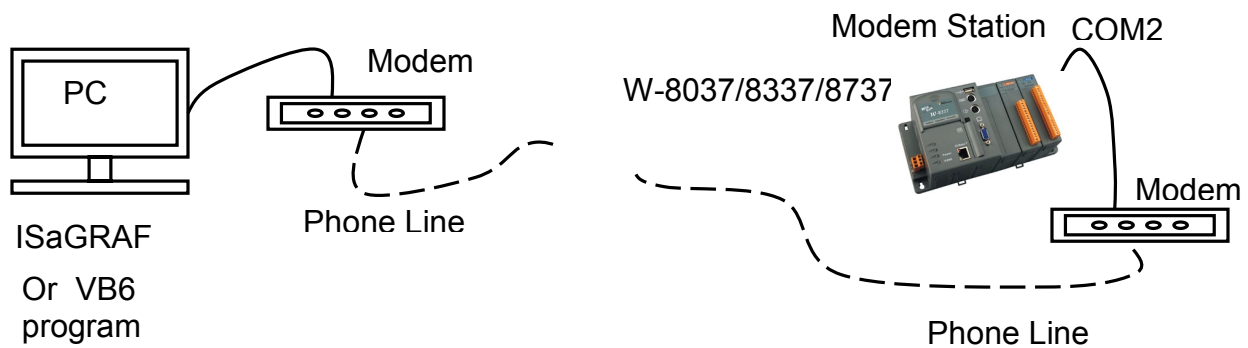


1.10: Modbus Master

COM2:RS232 or COM3:RS485 may link to other Modbus PLC or devices



1.11: Download & Monitoring Via Modem_Link



Chapter 2: Software Installation

Please refer to CD-ROM: \napdos\isagraf\wincon\english_manu\ "user_manual_i_8xx7.pdf" for detailed ISaGRAF English User's Manual.

The Wincon-8037/8337/8737 supports ISaGRAF programming method & provides Web HMI solution by default.

If user prefer to program Wincon-8037/8337/8737 by using Microsoft EVC++ 4.0 or VS.net 2003 (VB.net & C#.net) and access to the Web HMI, it is also possible. In such a case, user don't need to purchase any ISaGRAF Software (ISaGRAF-256-C or ISaGRAF-256-E). Please refer to Chapter 10.

If user would like to program W-8x37 by using both ISaGRAF & (EVC++ or VS.net), it is also possible. Please refer to Chapter 6 or Chapter 7.

2.1: Step 1 - Installing The ISaGRAF Software

The user has to install two kinds of software before he can program on the W-8037/8337/8737 controller system. They are

- A. **ISaGRAF Workbench** &
- B. **ICP DAS Utilities For ISaGRAF**

User has to purchase at least one pcs. of ISaGRAF (Ver. 3.4x, ISaGRAF-256-E or ISaGRAF-256-C) to install on his PC to edit, download, monitor & debug the controller system. Item (B) is free and it is burned inside the CD-ROM which is delivered with the W-8037/8337/8737.

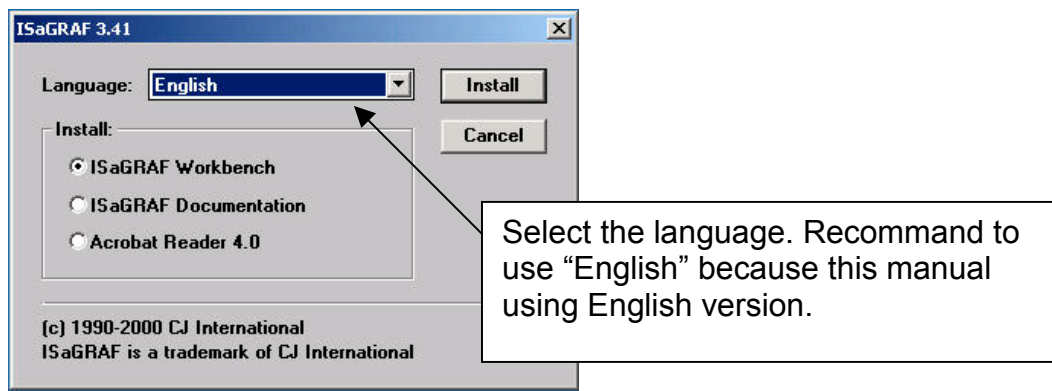
Operating system Requirements:

One of the following computer operating systems must be installed on the target computer system before you can install the ISaGRAF Workbench software program.

- Windows 95
- Windows 98
- Windows NT Version 3.51 or Windows NT Version 4.0
- Windows 2000 Or Windows XP

Steps To Installing The ISaGRAF Workbench:

Insert the ISaGRAF Workbench CD into your CD-ROM drive. If your computer does not have the auto-start feature active, use the Windows Explorer and go to the CD-ROM drive where the Workbench CD is installed, then double-click on the "install.bat" file listed on the ISaGRAF CD. If the "install.bat" file is not found on your ISaGRAF CD, then double-click on the "ISaGRAF.exe" file to start the installation process.

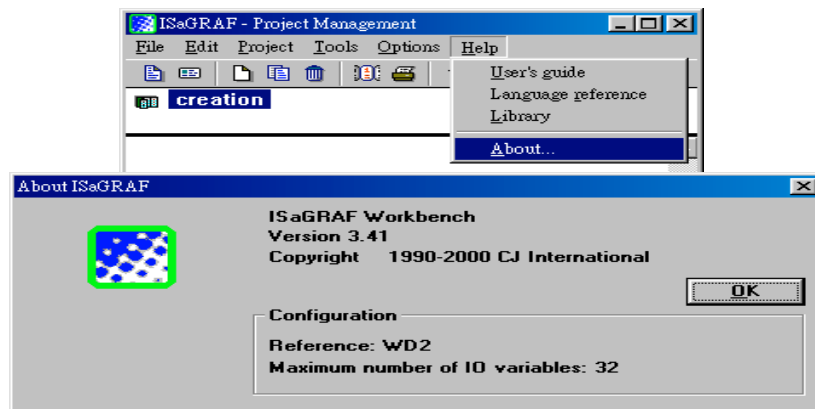


To begin the ISaGRAF 3.x software program, click on the Windows "Start" button, then on "Programs", and you should see the ISaGRAF program group as illustrated below.



NOTE: You must install the hardware protection device (dongle) provided with the ISaGRAF software on your computers parallel port to for the ISaGRAF program to achieve fully authorized functionality.

While using ISaGRAF and the dongle is plugged well, if the "Help" – "About" says "Maximum number of IO variables: 32", it means ISaGRAF workbench cannot find the dongle well. Please reset your PC and then check the "Help" – "About" again. If it still displays "Maximum number of



IO variables: 32", the dongle driver may not be installed well. Please execute the ISaGRAF CD_ROM \Sentinel5382\setup.exe for ISaGRAF-80 or \Sentinel\setup.exe for other ISaGRAF version and then reset the PC again.

Important Notice For Window NT Users

If your computer is using the Windows NT operating system, you will need to add one line to the "isa.ini" file in the ISaGRAF Workbench "EXE" subdirectory.

C:\isawin\exe\isa.ini

You can use any ASCII based text editor (such as Notepad or UltraEdit32) to open the "isa.ini" file. Locate the [WS001] header in the "isa.ini" initialization file (it should be at the top of the file). Anywhere within the [WS001] header portion of the "isa.ini" initialization file, add the entry shown below within the [WS001] header:

[WS001]

NT=1

Isa=C:\ISAWIN

IsaExe=C:\ISAWIN\EXE

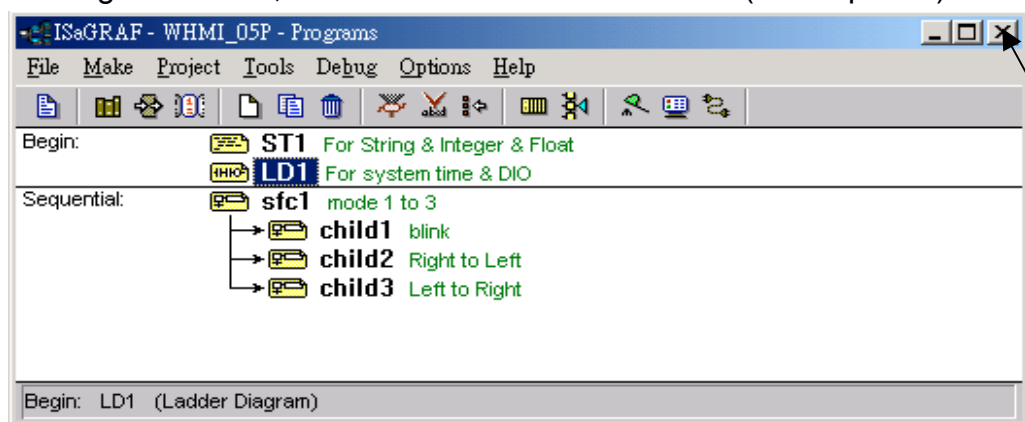
Group=Samples

IsaApl=c:\isawin\smp

IsaTmp=C:\ISAWIN\TMP

2.1.1: When closing my ISaGRAF window on windows 2000, it holds. Why ?

This problem usually happens on the windows 2000. When you close some ISaGRAF windows by clicking on the "X", it holds about 20 to 40 seconds (No response).



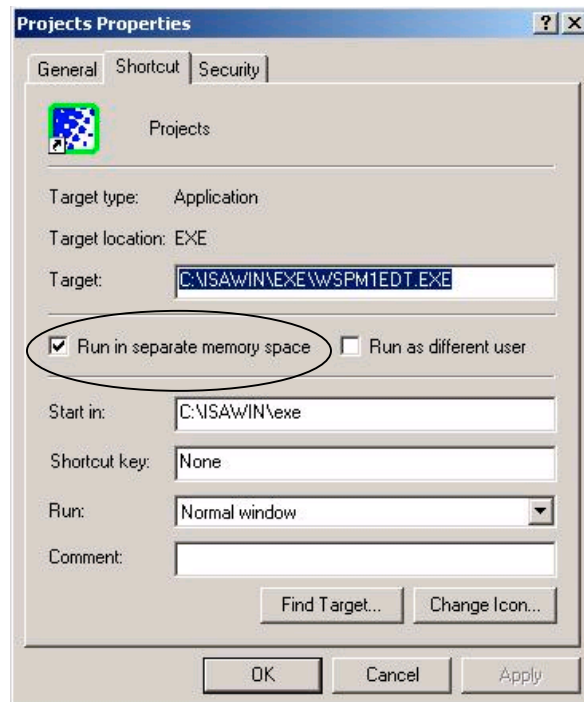
This "hold" behavior is caused by the "CTFMON.EXE" process. We still don't know the reason yet. You may stop this process by click on the "Ctrl" & "Alt" & "Del" at the same time to open the window Task Manager, and then stop it as next page.

However you will find the "CTFMON.EXE" still load to run when you reboot your PC or run Microsoft Office. So you need to stop it every time when your windows 2000 is rebooted. If you want to know more about the "CTFMON.EXE", please visit www.microsoft.com & search "CTFMON.EXE".



2.1.2: One Quick way to avoid the “hold” problem on windows 2000.

You may create a short cut for the “ISaGRAF project manager. And then check on "run in separate memory space" option in the shortcut property.



2.2: Step 2 - Installing The ICP DAS Utilities For ISaGRAF

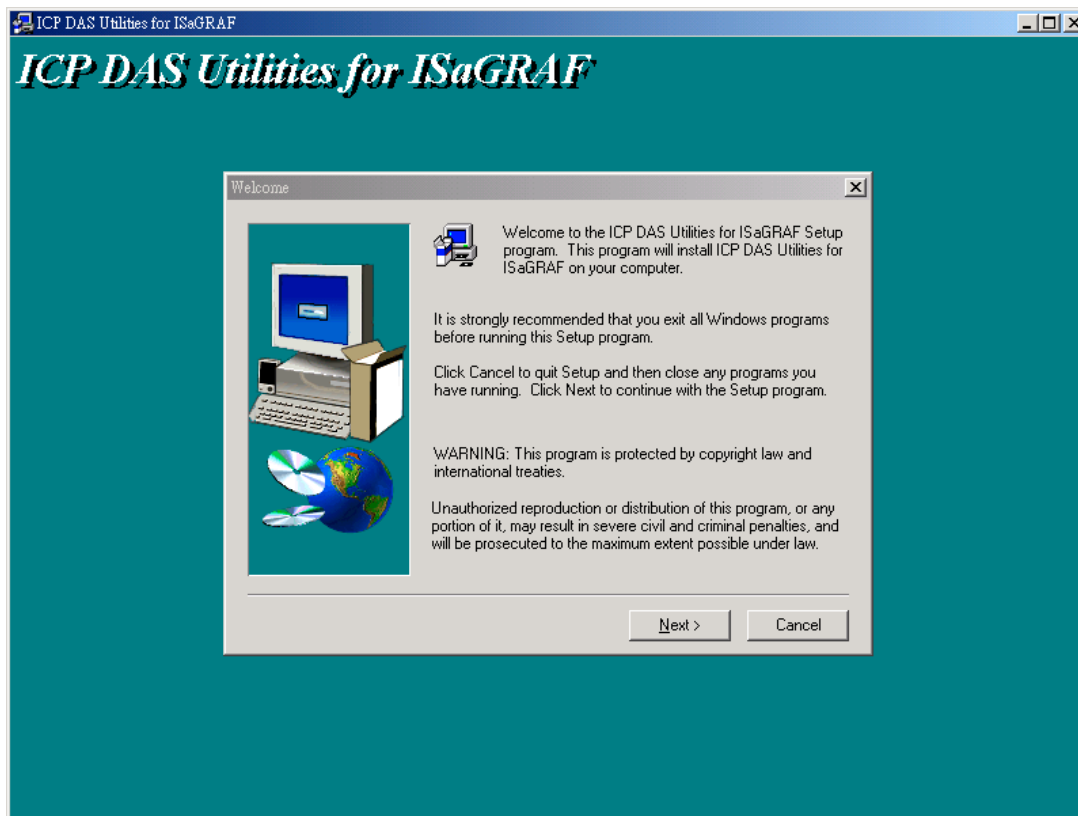
The “ICP DAS Utilities For ISaGRAF” consists of 3 major items.

I/O libraries of I-8417/8817/8437/8837, I-7188EG, I-7188XG & W-8037/8337/8737
Modem_Link utility
Auto-scan I/O utility

Note:

The ISaGRAF Workbench software program must be installed before attempting to install the “ICP DAS Utilities for ISaGRAF”. If you have not already installed the ISaGRAF Workbench program, please refer to **step 1** before continuing.

There is a CD-ROM supplied with each of the W-8037/8337/8737 controllers with the “ICP DAS Utilities for ISaGRAF”. Please insert the CD-ROM into your CD-ROM drive. Then run **CD-ROM: \napdos\isagraf\setup.exe** . Follow the steps to install it.



Note:

If “ICP DAS Utilities for ISaGRAF” is not in your CD-ROM, please download “**ICP DAS Utilities For ISaGRAF.zip**” from <http://www.icpdas.com/products/PAC/i-8000/isagraf.htm>.

2.3: Step 3 - Installing The Web Page Editor

This is an option. You may not need it if you are very familiar with the HTML design. It is also possible to use any text editor to build web pages. For example, “Notepad” on the windows 2000 or XP.

We will use “Microsoft Office FrontPage 2003” (or higher version) to build web pages in this manual.

User may choose your prefer web page editor to do the same thing.

Chapter 3: Setting Up A Web HMI Demo

This Chapter lists steps to set up a Web HMI Demo in the Wincon-8037/8337/8737.

3.1: Web Demo List

The Web page location:

Wincon-8xx7 CD-ROM: \napdos\isagraf\wincon\WebHMI_Demo\

The respective ISaGRAF project location:

Wincon-8xx7 CD-ROM: \napdos\isagraf\wincon\demo\

New demo will be found at <http://www.icpdas.com/products/PAC/i-8000/isagraf.htm>

Demo list:

Name	Description	IO board
sample	A Web HMI sample	No I/O board
example1	A simple example listed in Chapter 4	slot 1: I-8077
whmi_01	Display controller's date & time	No I/O board
whmi_02	DI & DO demo	slot 1: I-8077
whmi_03	Read / Write Long, float & Timer value	No I/O board
whmi_04	Read / Write controller's String	No I/O board
whmi_05	Multi-Pages demo (slot 1: I-8077) Page menu is on the Left	slot 1: I-8077
whmi_05a	Multi-Pages demo (slot 1: I-8077) Page menu is on the Top	slot 1: I-8077
whmi_06	AIO demo, scaling is in ISaGRAF	slot 2: I-87024 slot 3: I-8017H
whmi_07	AIO demo, scaling is in PC	slot 2: I-87024 slot 3: I-8017H
whmi_08	download controller's file to PC	slot 1: I-8077
whmi_09	pop up an alarm window on PC	slot 1: I-8077
whmi_10	Entering data to the controller and then store them into file.	slot 1: I-8077

3.2: Steps To Set Up A Web HMI Demo

3.2.1: Step 1 - Setup The Hardware

A. Please have a Wincon-8337 or Wincon-8737 & plug one I-8077 board in its slot 1.

If you don't have the I-8077 (8 IN & 8 OUT simulation board), please follow the same steps as below however your Web HMI demo may be replaced to "whmi_01" not "whmi_05"

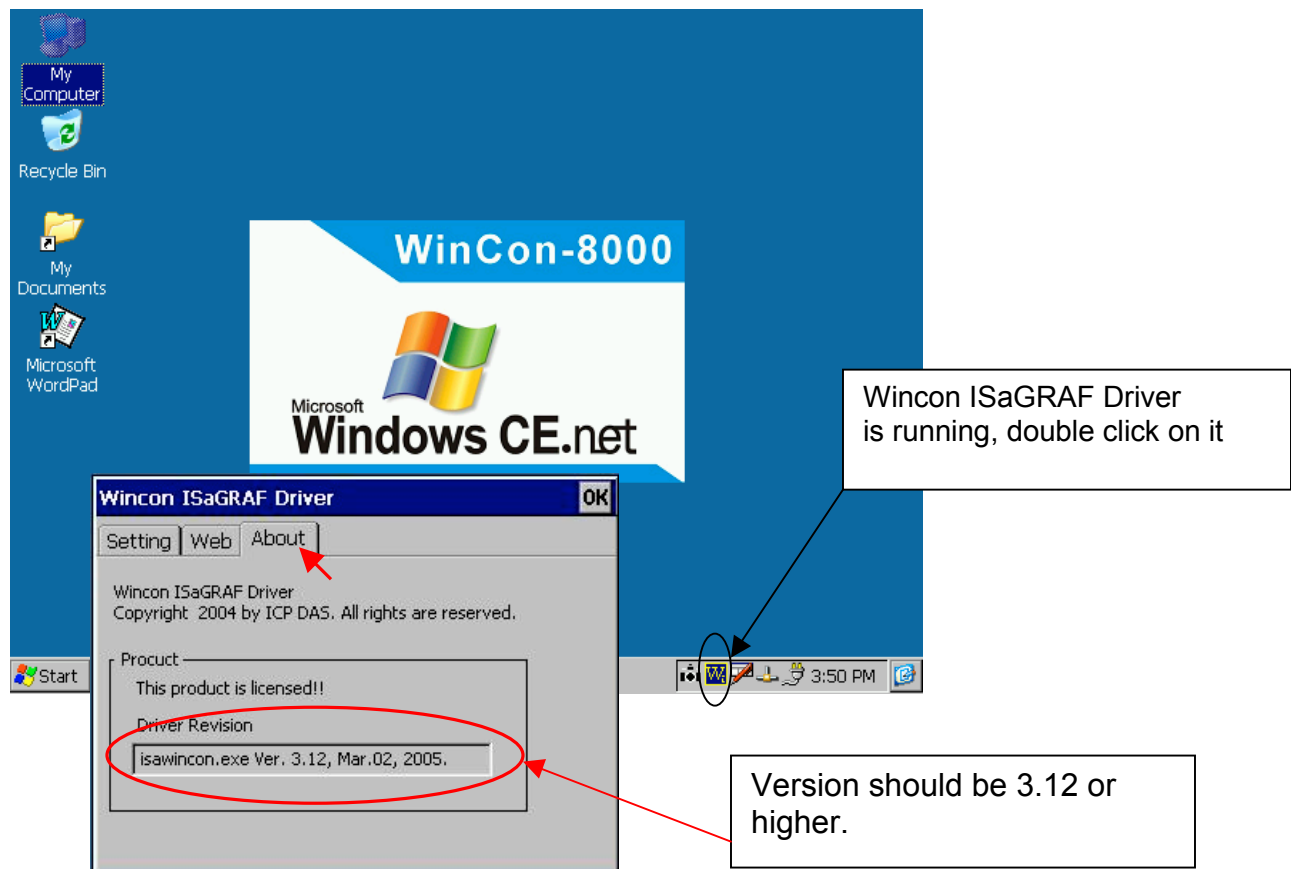
B. Prepare one VGA monitor, one PS2 mouse, one Keyboard and one ethernet cable and then connect them to the Wincon.

C. Power the Wincon up.

3.2.2: Step 2 - Make Sure The Right ISaGRAF Driver Installed

To run a Web HMI Demo, please make sure the ISaGRAF driver installed in the Wincon-8037/8337/8737 is version 3.12 or high. If the version is lower than 3.12, please upgrade it. The procedure is listed in Appendix B or in the W-8xx7 CD-ROM:
\\napdos\isagraf\wincon\english_manu\ "Upgrade_w8337.pdf"

Note: If you have upgraded the driver, please recycle the power of your W-8037/8337/8737.

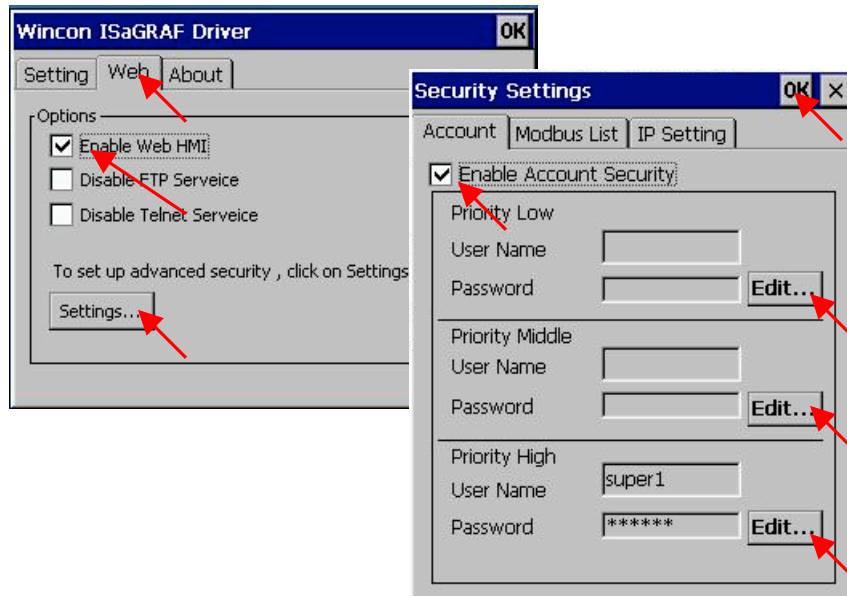


3.2.3: Step 3 - Setting The Web Options

A. Please set a fixed IP address to the Wincon-8037/8337/8737. (No more DHCP)

B. Check on “Enable Web HMI” and then click on “Setting”, Please check on “Enable Account Security” and then click on “Edit” to set (username , password). **Then remember to click on “OK”**

Note: If “Enable Account Security” is not check, any user can easily get access to your Wincon through the Internet Explorer.



3.2.4: Step 4 - Download ISaGRAF Project

Please download ISaGRAF project “whmi_05” to the W-8337/8737. This project is in the CD-ROM:\napdos\isagraf\wincon\demo\ “whmi_05.pia”

whmi_05 demo need one I-8077.If you don't have the I-8077 (8 IN & 8 OUT simulation board), you may download “whmi_01” (CD-ROM:\napdos\isagraf\wincon\demo\ “whmi_01.pia”)

If you know how to restore “whmi_05.pia” to your ISaGRAF Workbench and download it to the controller, please go ahead to section 3.2.5.

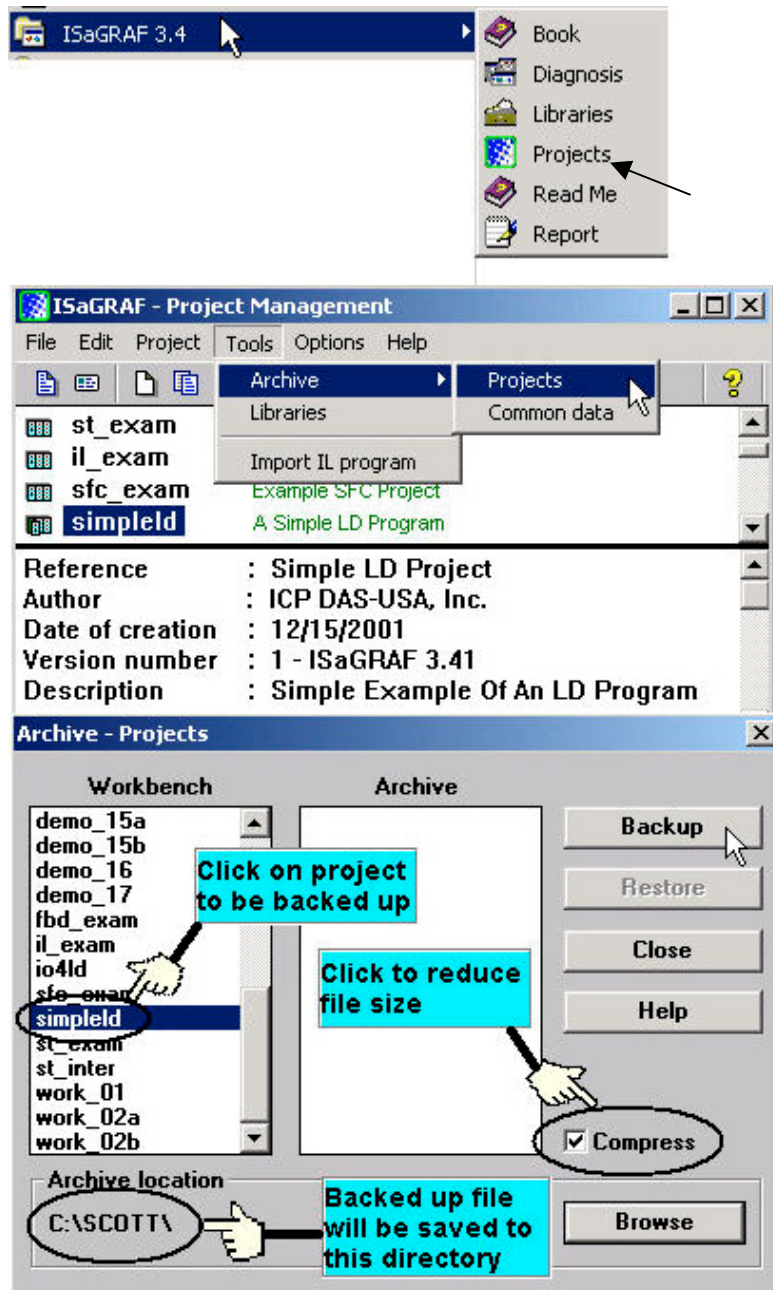
However if you don't know it, please refer to the below steps. Please make sure the ISaGRAF Workbench is already installed to your PC. (refer to section 2.1 & 2.2)

Steps To Backing Up & Restoring An ISaGRAF Project:

For archiving purposes you can "Back Up" and "Restore" an ISaGRAF project. For example, you may want someone to test your program or email to service@icpdas.com for ICP DAS's ISaGRAF technical service.

Backing Up An ISaGRAF Project

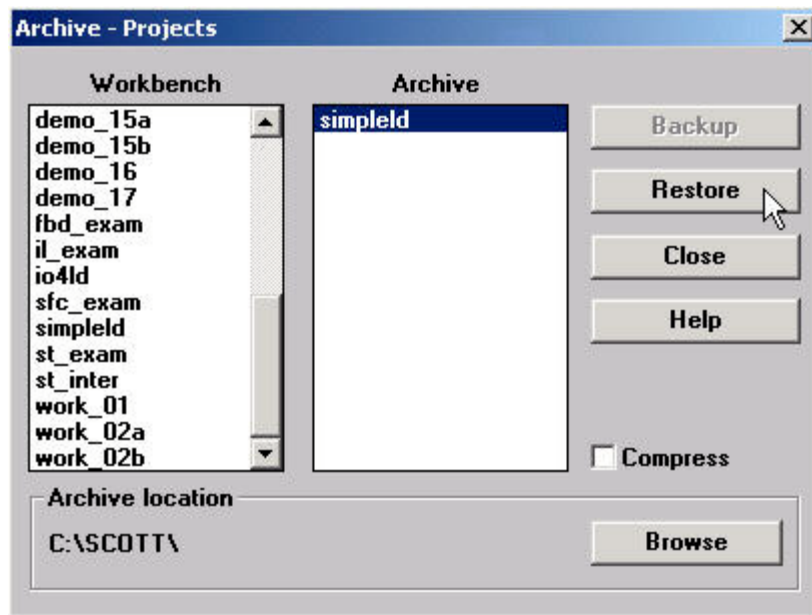
Open the "ISaGRAF Project Management" , select "Tools" from the menu bar, click on "Archive", and then click on "Projects". An "Archive Projects" window will open which allows you to designate where you want to save the ISaGRAF project to. Click on the name of the ISaGRAF project you want to backup, and then click on the "Backup" button. You can compress the size of the file you have backed up by clicking on the "Compress" checkbox BEFORE you click on the "Backup" button.



You will now find the backed up ISaGRAF project file in the "Archive" location you have designated. In the example above, the name of the backed up file is "simpleld.pia".

Restoring An ISaGRAF Project

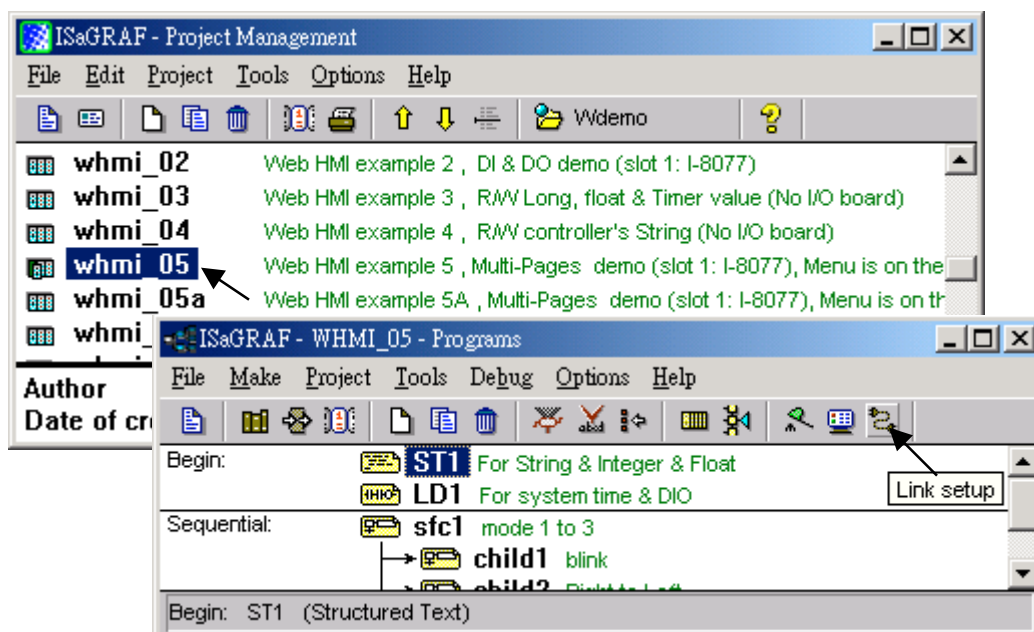
To restore an ISaGRAF project from a backed up file(*.pia), use the same method as above to access the "Archive Projects" window, click on the name of the project you want to restore from the "Workbench" window, then click on the name of the backed up file from the "Archive" window, then click on the "Restore" button. The ISaGRAF project will now be restored to the sub-directory you designated.



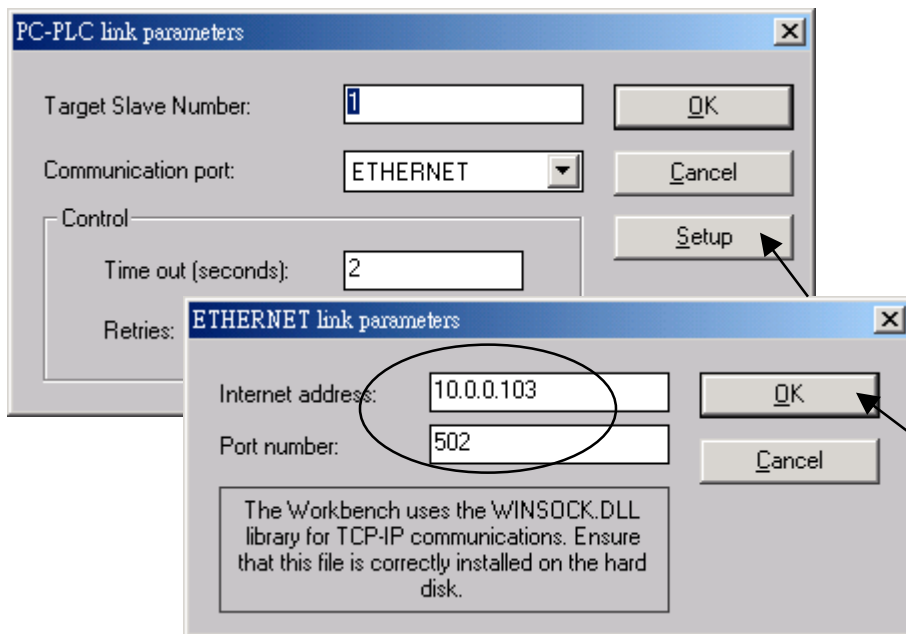
You can now open, edit and download the restored ISaGRAF project file.

3.2.4.1: Steps To Download a ISaGRAF Project To The Controller:

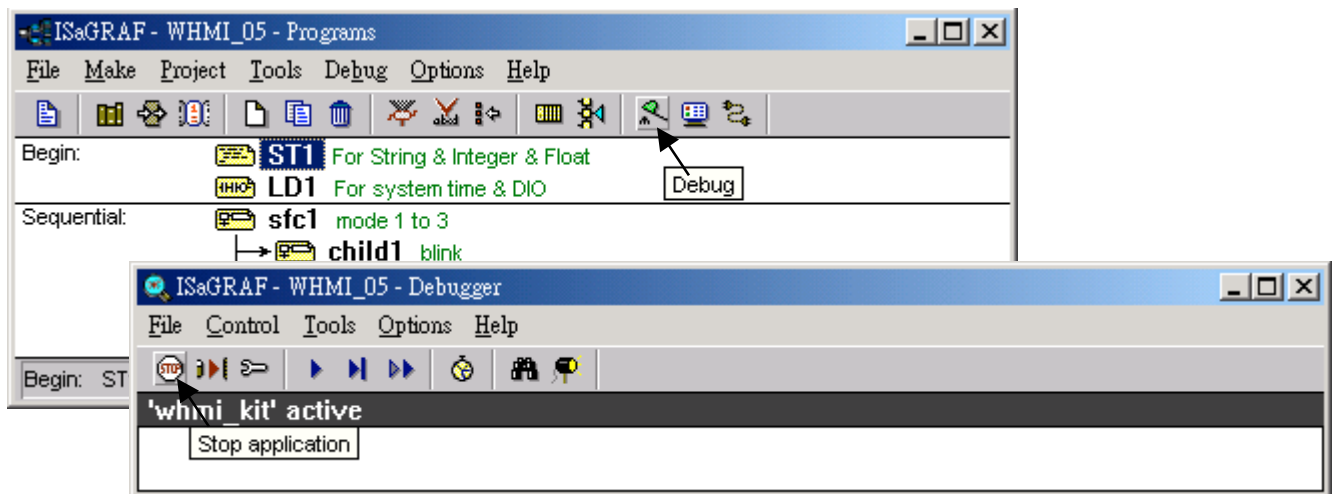
Double click on the "whmi_05" to get into the project. Then click on "Link setup".



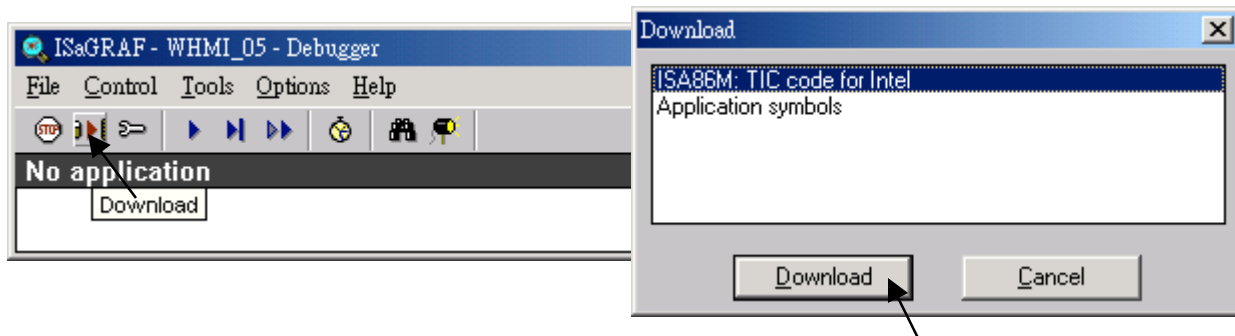
Click on “Setup” first and then entering the IP address of your controller. The port number should be 502.



To download “whmi_05” project to the Wincon-8xx7, Click on “Debug” . If the communication is established, click on “stop” first to stop the old project running in the W-8xx7.



Then click on “Download” to download it to the controller.

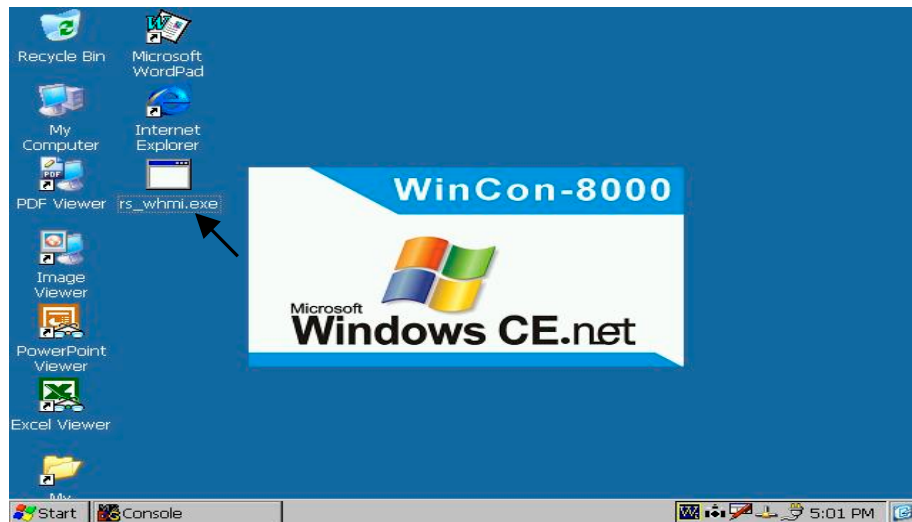


3.2.5: Step 5 - Download Web Pages To The Wincon

A. Please copy all files in the CD-ROM:\napdos\isagraf\wincon\WebHMI_demo\whmi_05\ *.* to the W-8037/8337/8737's Compact Flash\Temp\HTTP\WebHMI\

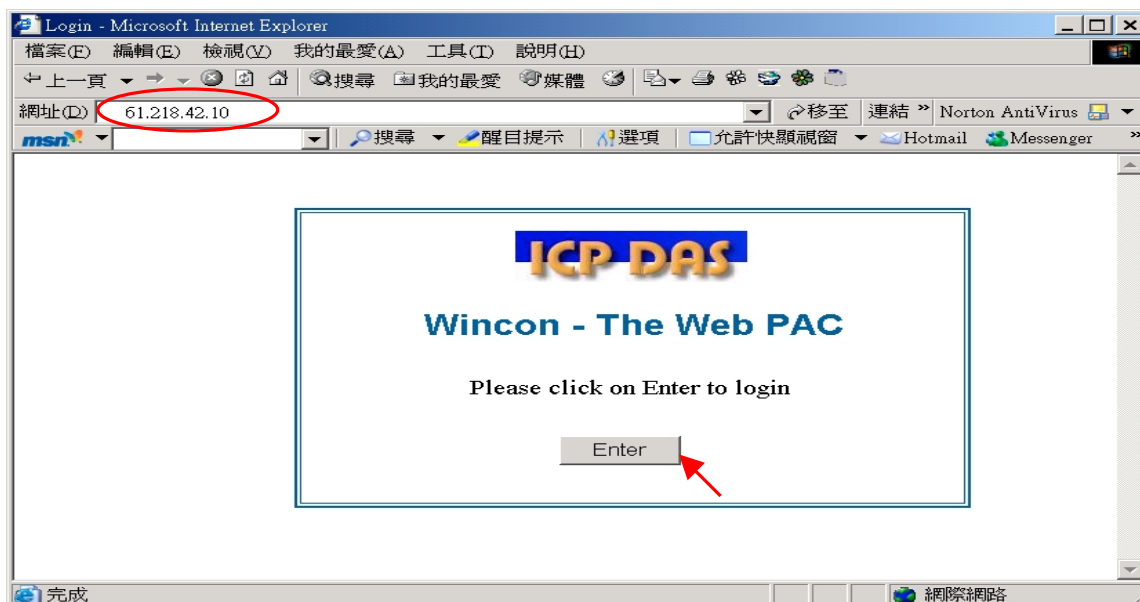
whmi_05 demo need one I-8077.If you don't have the I-8077 (8 IN & 8 OUT simulation board), you may download "whmi_01" (CD-ROM:\napdos\isagraf\wincon\WebHMI_demo\whmi_01\)

B. Since the Web Pages are modified or new copied, please run "rs_whmi.exe" to reset the Web server. **The "rs_whmi.exe" must be run every time when user has modified any file in the W-8x37's Compact Flash\Temp\HTTP\WebHMI**



3.2.6: Step 6 - Show Time

Please run Internet Explorer (Rev. 6.0 or higher), key in the IP address of your W-8037/8337/8737. For example: 61.218.42.10 or http://61.218.42.10



Chapter 4: Programming A Web HMI Example

This chapter show you how to build a simple ISaGRAF project and its Web HMI pages.

Please refer to CD-ROM: \napdos\isagraf\wincon\english_manu\ "user_manual_i_8xx7.pdf" - Section 2.1 for detailed ISaGRAF programming basics.

The Wincon-8037/8337/8737 supports ISaGRAF programming method & provides Web HMI solution by default.

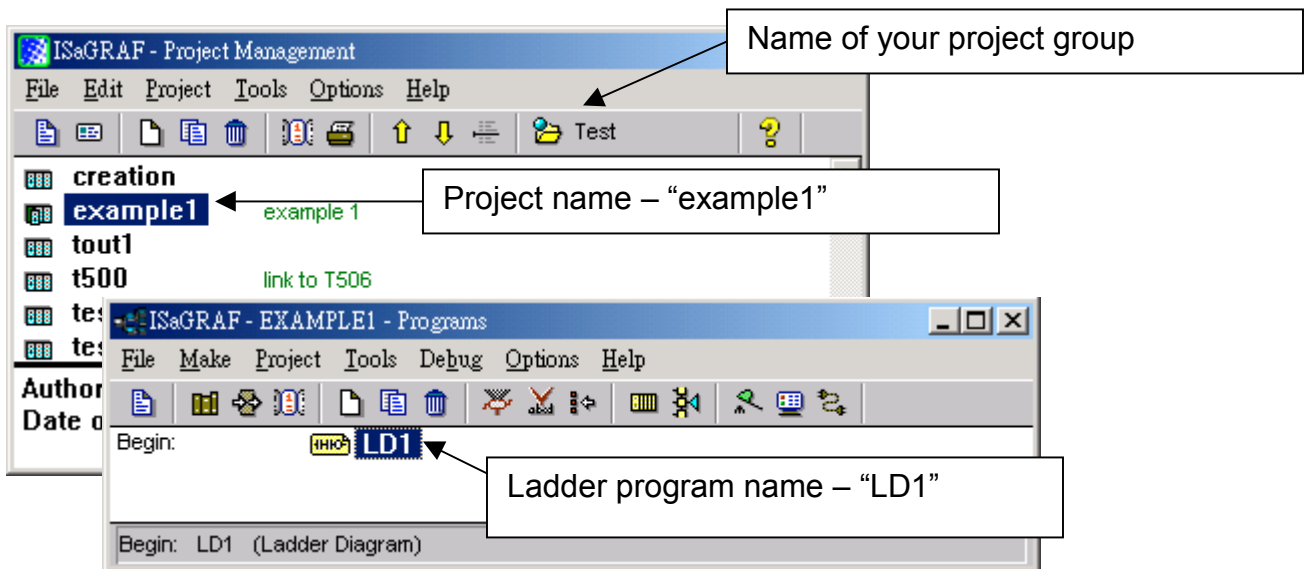
If user prefer to program Wincon-8037/8337/8737 by using Microsoft EVC++ 4.0 or VS.net 2003 (VB.net & C#.net) and access to the Web HMI, it is also possible. In such a case, user don't need to purchase any ISaGRAF Software(ISaGRAF-256-C or ISaGRAF-256-E). Please refer to Chapter 10.

If user would like to program W-8x37 by using both ISaGRAF & (EVC++ or VS.net), it is also possible. Please refer to Chapter 6 or Chapter 7.

4.1: Writing A Simple ISaGRAF Program

We are going to use ISaGRAF Workbench to write a simple ISaGRAF example program, then download it to the W-8037/8337/8737 controller (with one **I-8077** I/O board in it) to make it work. If you haven't installed "ISaGRAF" & "ICP DAS Utilities for ISaGRAF", please back to chapter 2.

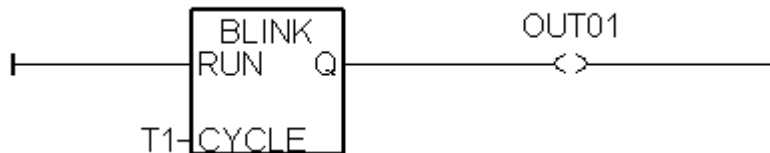
This example contains one Ladder program.



Variables declaration:

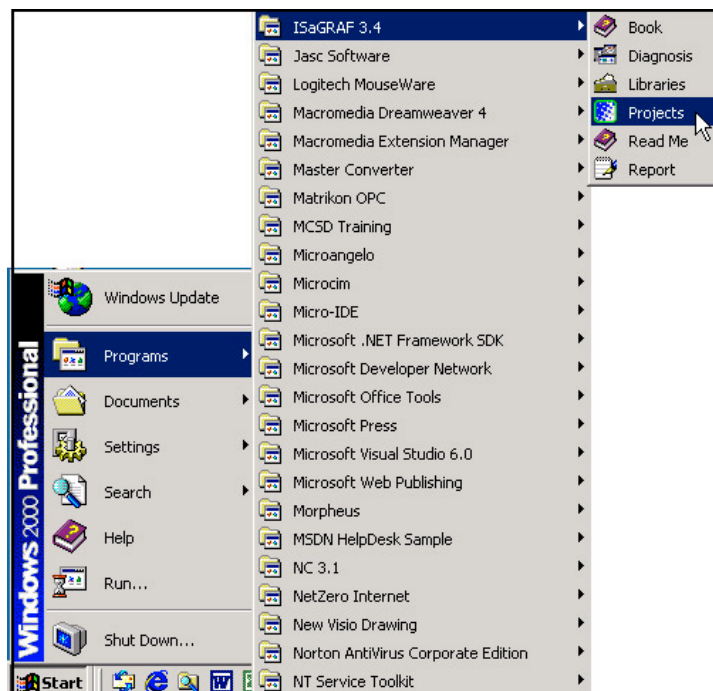
Name	Type	Attribute	Description
OUT01	Boolean	Output	Output 1 in the I-8077 board, Modbus network addr = 1
OUT02	Boolean	Output	Output 2 in the I-8077 board, Modbus network addr = 2
K1	Boolean	Input	Input 1 in the I-8077 board, Modbus network addr = 11
K2	Boolean	Input	Input 2 in the I-8077 board, Modbus network addr = 12
T1	Timer	Internal	Time Period of blinking, initial value set as T#8s Modbus network addr = 21

Ladder Logic Program Outline:



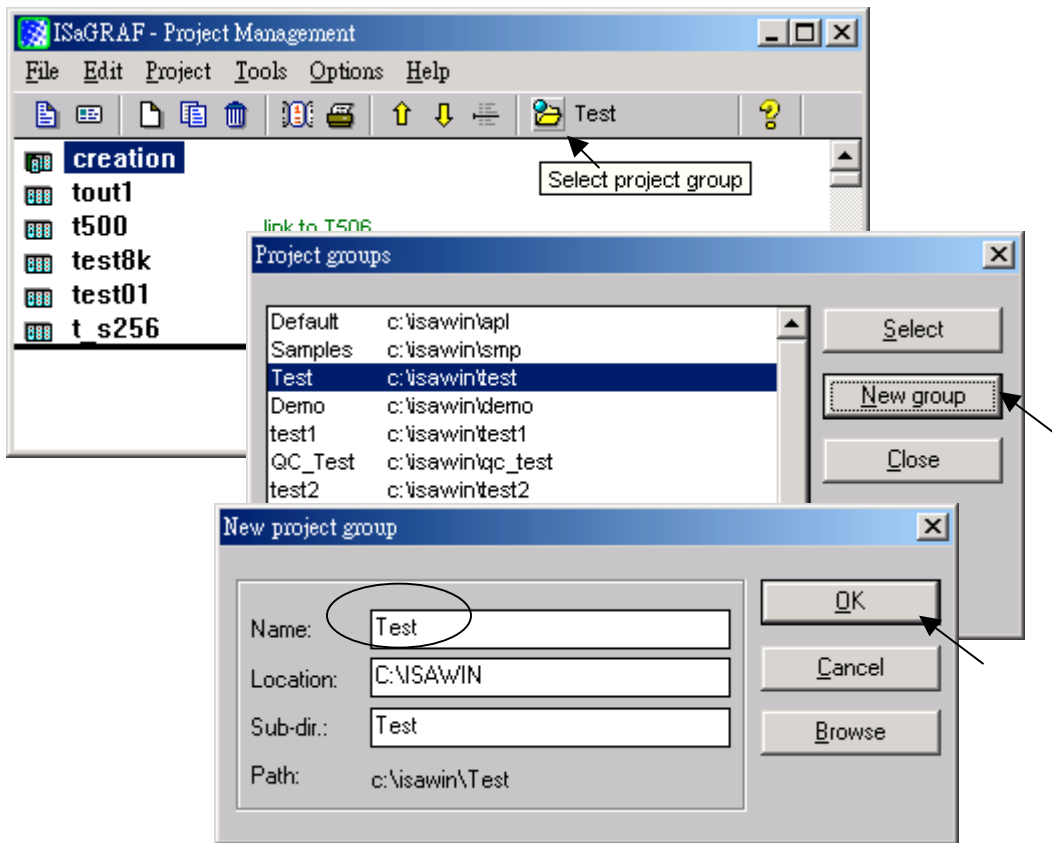
4.1.1: Open ISaGRAF-Project Management

Click on the Windows "Start" button, then click on "Programs", then click on "ISaGRAF 3.4", then click on "Projects" as shown below.



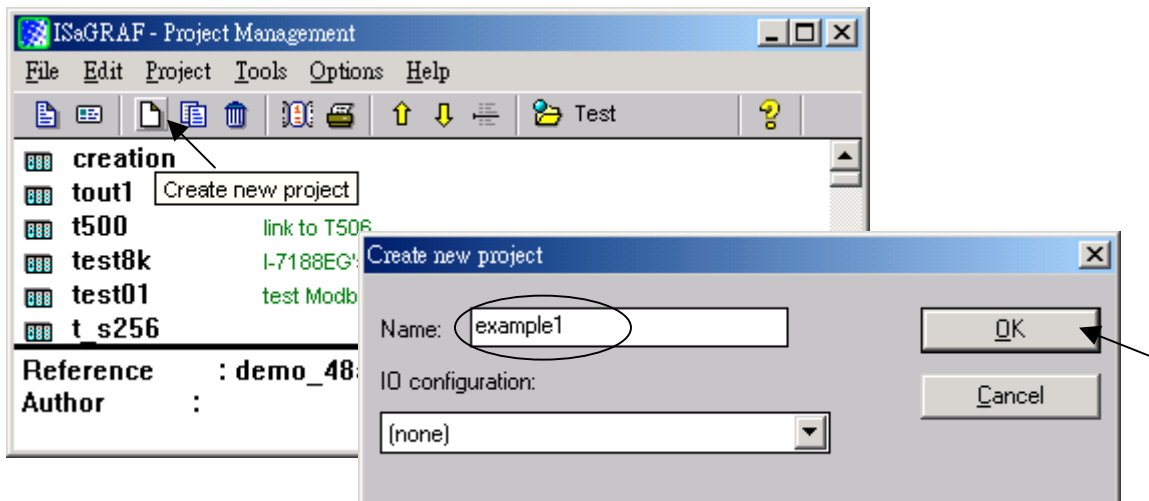
4.1.2: Creating An ISaGRAF User's Group

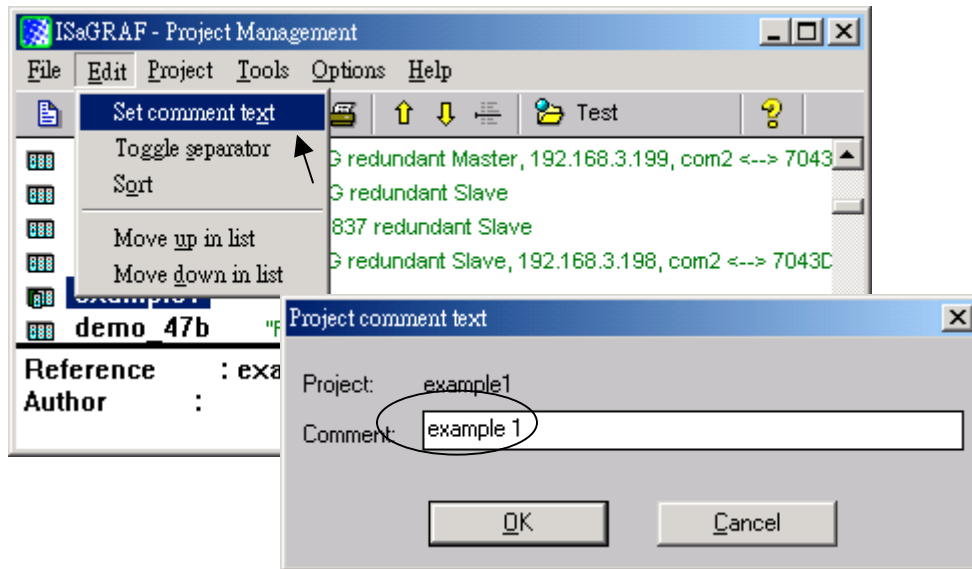
Click on the "Select Project Group", and then click on "New Group", then type in the name for the new user's group you wish to create, and last click on "OK".



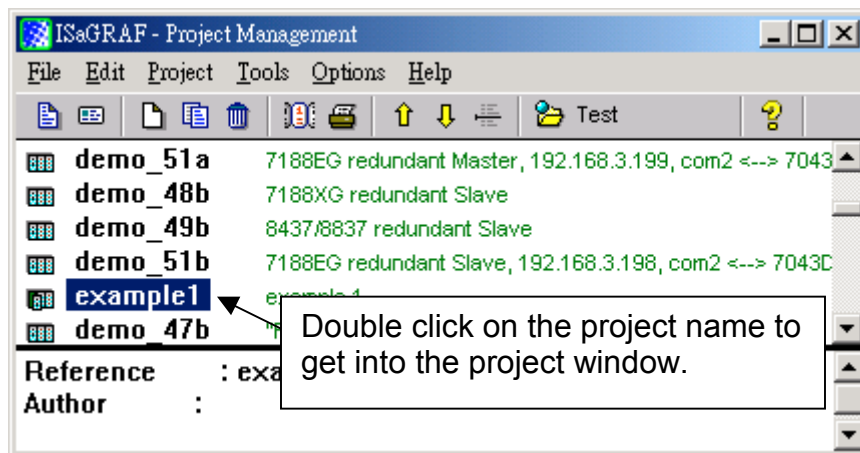
4.1.3: Creating A New ISaGRAF Project

To start a new ISaGRAF project, click on the "Create New Project" icon and then enter in the name for the new project. You can then enter additional information for your project by clicking on the "Edit" and then "Set Comment Text" menu as illustrated below.



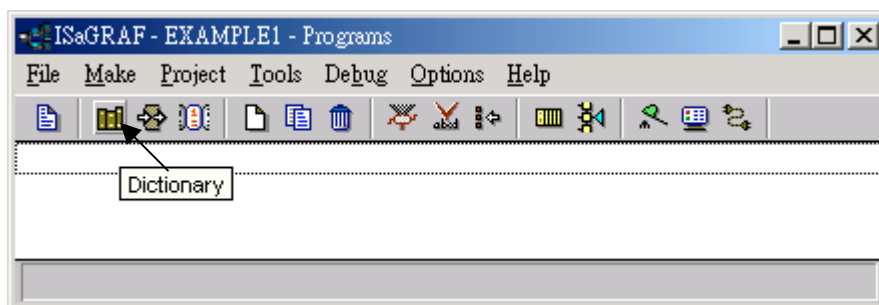


You will now see the name of the new project in the "Project Management" window. Double click on the name of the new project to open the new project.



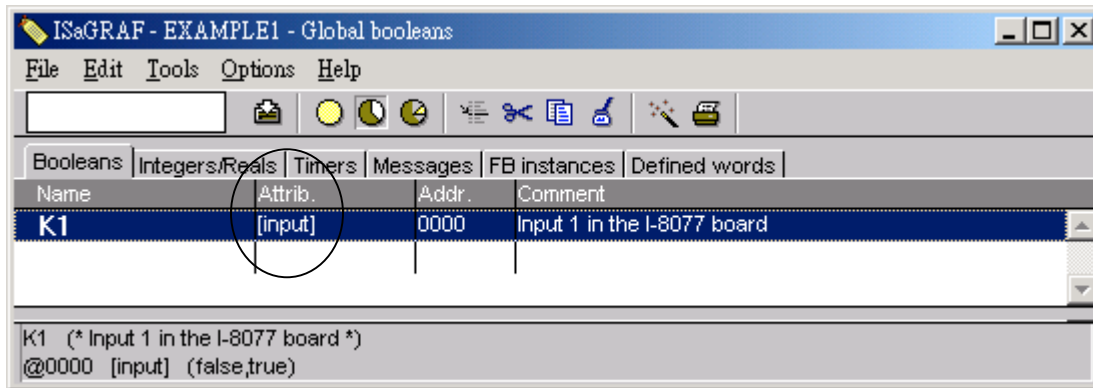
4.1.4: Declaring The ISaGRAF Project Variables

Before you can start creating an ISaGRAF program, you must first declare the variables that will be used in the ISaGRAF program. To begin this process, first click on the "Dictionary" icon and then click on the "Boolean" tab to declare the Boolean variables that will be used in our example program.



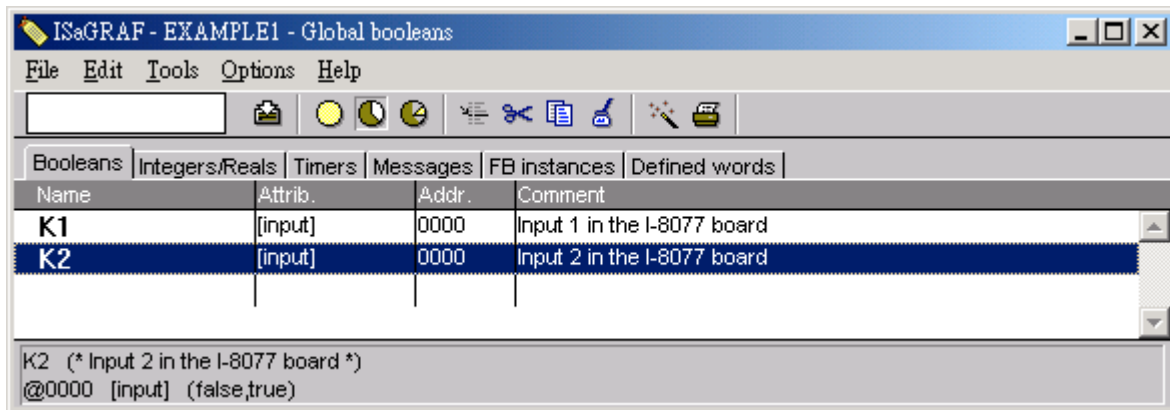
To declare the program variables for the ISaGRAF project, double click on the colored area below the "Boolean" tab, and a "Boolean Variable" window will open. Enter in the name of the variable to be used in the project. For the purpose of this example program the variable "Boolean Variable Name" is "K1", and "Input 1 in the I-8077 board" is added to the "Comment Section". The next item that must be declared is what type of "Attribute" the variable will possess. In this example program, K1's attribute will be an "Input". Then press the "Store" button to save it.

The new Boolean variable has now been declared.

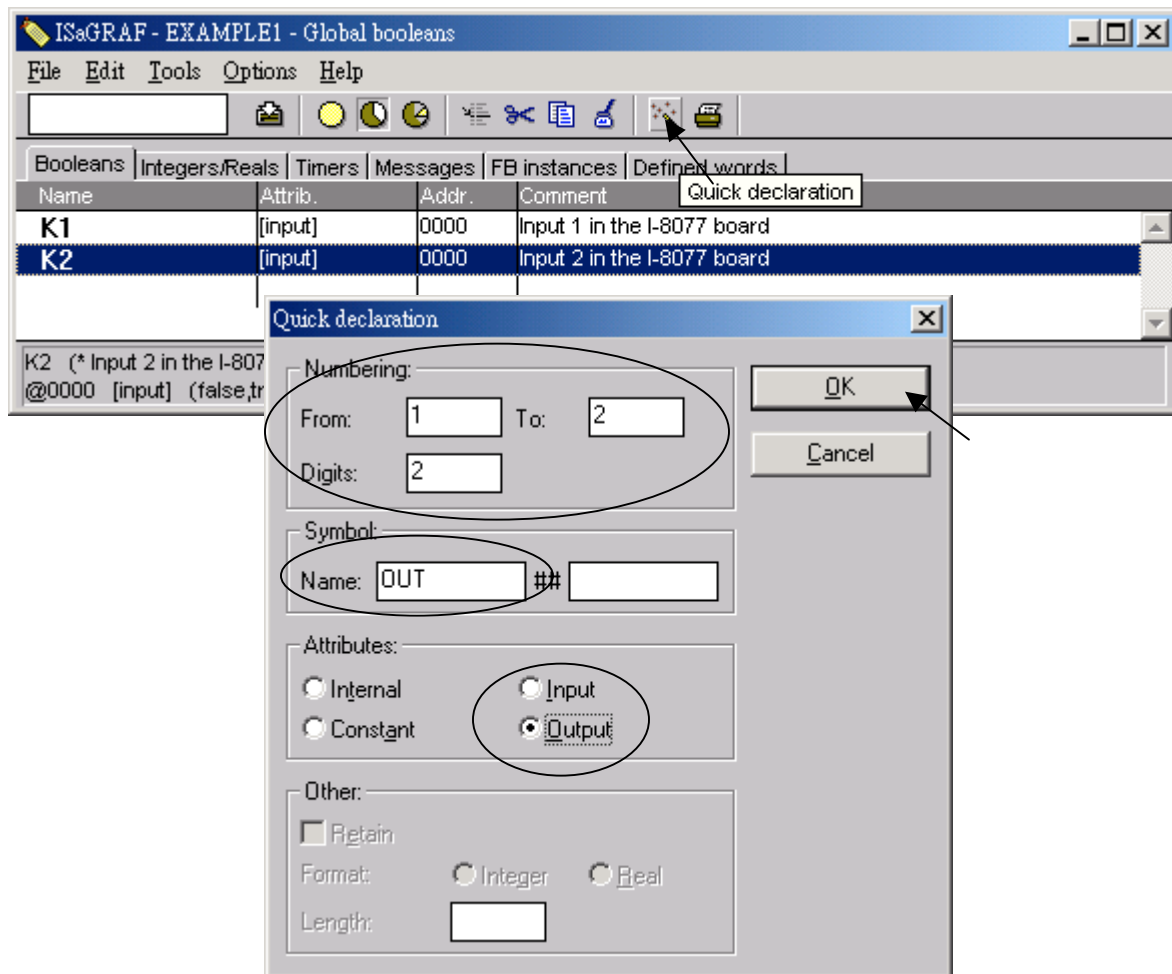


NOTE: You MUST make sure that the variable you have declared has the desired **Attribute** assigned. If you decide that you want to change a project variable's attribute, just double click on the variable name and you can reassign the attribute for the variable

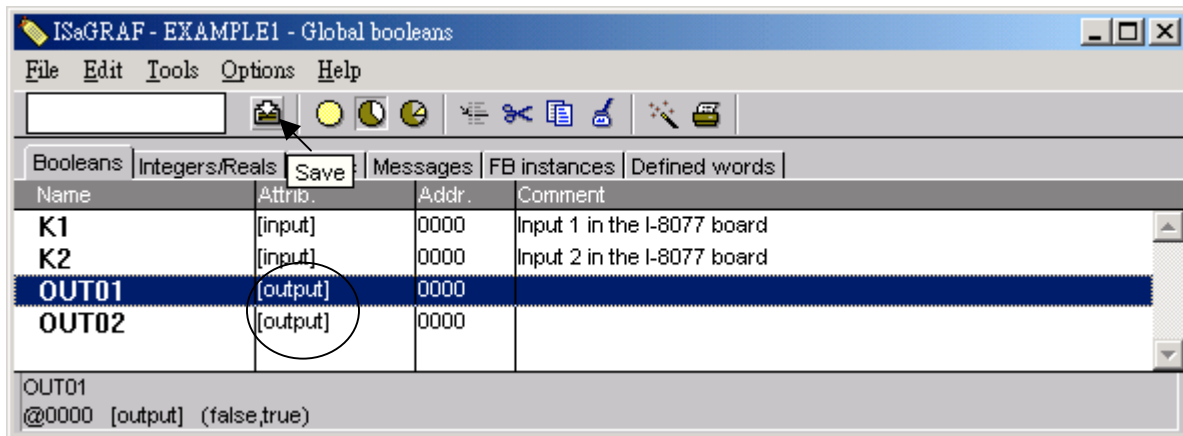
Please follow the above same step to declare one another boolean variable – “K2”. Then you will have as below.



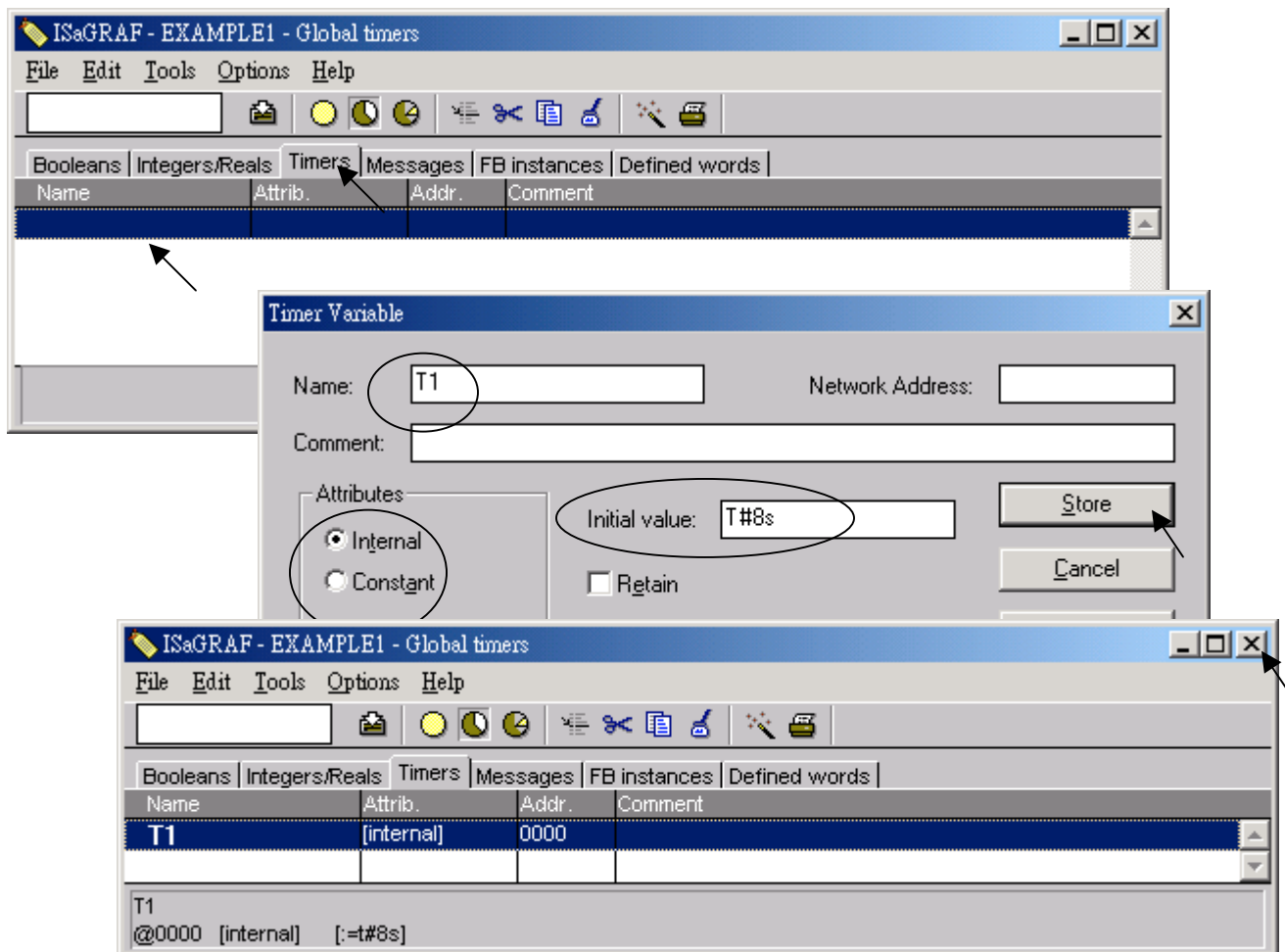
There are two outputs used in this example program named "OUT01 and OUT02". ISaGRAF provides a quick and easy way to declare like variables that are sequentially ordered. To begin this process, click on the "Quick Declaration" icon, and enter in the output number that you will start with in the "Numbering" from and "To" field (this example uses from 1 to 2). Enter the "Symbol" name for the output variables being declared, and lastly, set the attribute to "Output"



When you click on the "OK" button, all two outputs will be immediately added to the "Global Boolean" window. Click on Save to store them.



To declare the timer (T1) variable used in this example program, click on the "Timers" tab in the setup screen. Double click on the colored area and enter the Name as "T1", set the "Attributes" to "Internal", the "Initial Value" to "T#8s", then click on the "Store" button. Then please click on "X" to close the "dictionary" window.



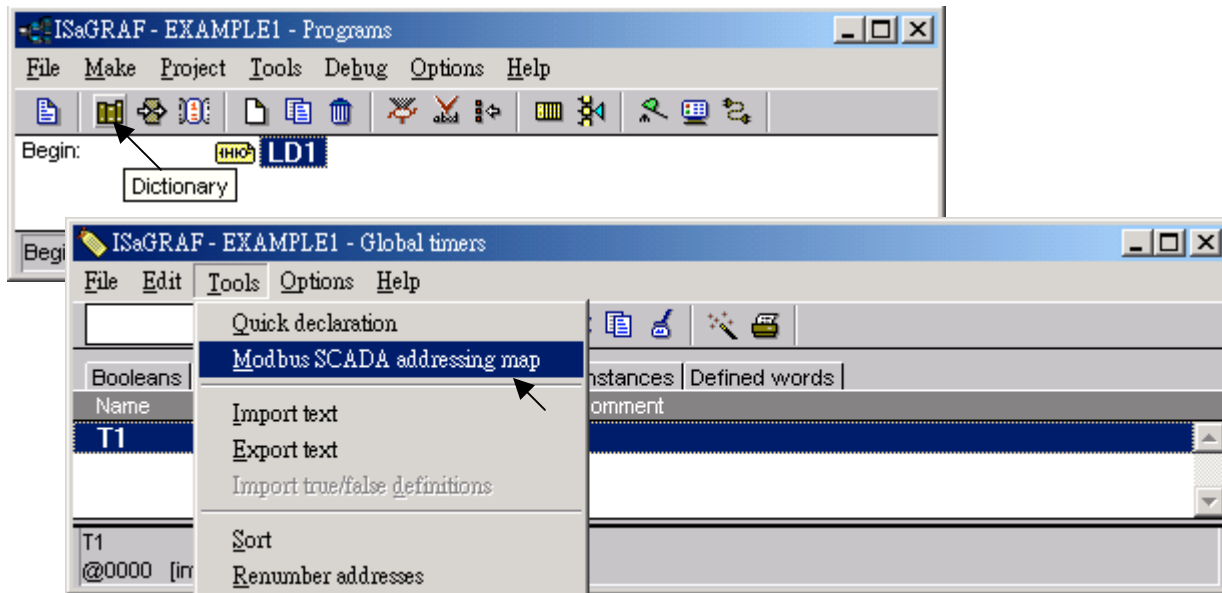
4.1.5: Assign Modbus Network Address No to Variables

The Web HMI will exchange the variable value with the ISaGRAF project if they have assigned the proper “Modbus network address”. The Web HMI only recognize Modbus No. from 1 to 1024. However other SCADA software may R/W the Modbus No. from 1 to 8191 in the W-8xx7.

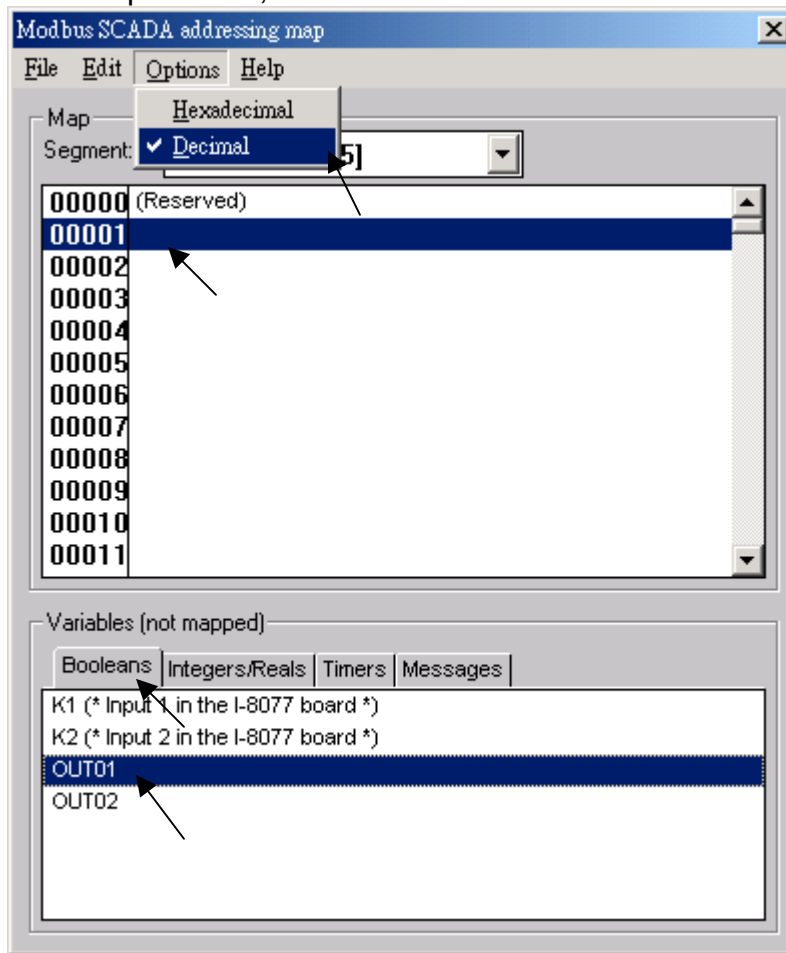
Variables without assigning Modbus No. will not be available by Web HMI and other SCADA software or HMI devices.

Please refer to CD-ROM: \napdos\isagraf\wincon\english_manu\ "user_manual_i_8xx7.pdf" - Section 4.1 & 4.2 for detailed information about assigning Modbus network address.

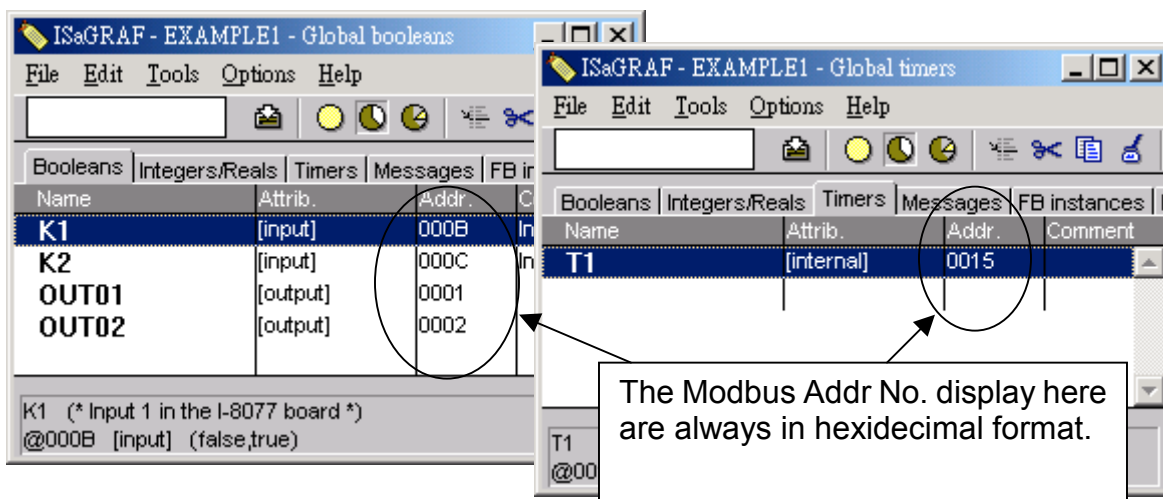
Please get into the dictionary, then click on “Tools – Modbus SCADA addressing map”



Please click on “Options – Decimal” , or it will use Hexadecimal format as default. First click on “00001” on the top window, and then double click on “OUT01” to attach it to the Modbus No. 1.



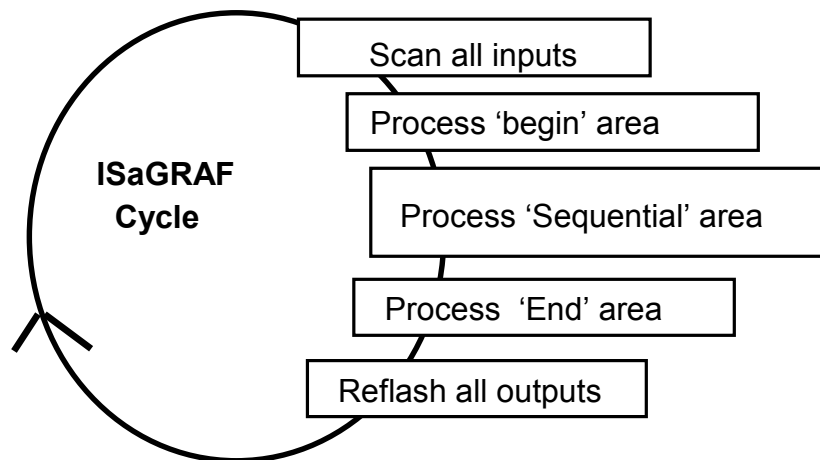
Please follow the same method to assign OUT01 to No.2 , K1 to No.11 , K2 to No.12 and then Timer variable T1 to No.21 . Then we have below window.



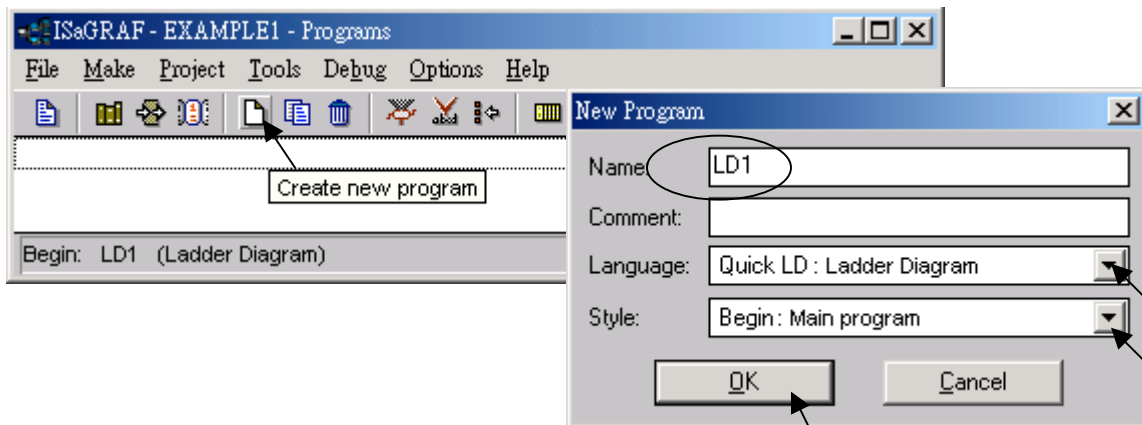
Very Important: If assign Modbus No. to Long integer or Float or Timer variables, they should occupy two Modbus No. Please refer to CD-ROM: \napdos\isagraf\wincon\english_manu\ "user_manual_i_8xx7.pdf" - Section 4.2 for detailed information.

4.1.6: Create The LD - "LD1" Program

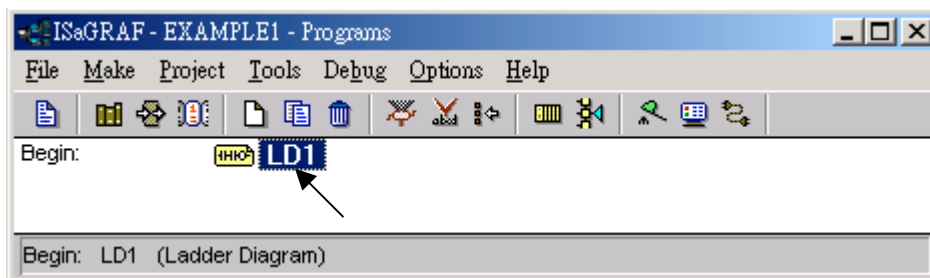
ISaGRAF will run every program one time in each PLC scan cycle. Programs in the "begin" area will run first, then the "Sequential" area, and last the "End" area. An ISaGRAF cycle run in the way as the below scheme.



Click on the "Create New Program" icon and the "New Program" window will appear. Enter the "Name" as "LD1", next, click on the "Language" scroll button and select "Quick LD: Ladder Diagram", and make sure the "Style" is set to "Begin: Main Program". You can add any desired text to the "Comment" section for the LD program, but it isn't required.

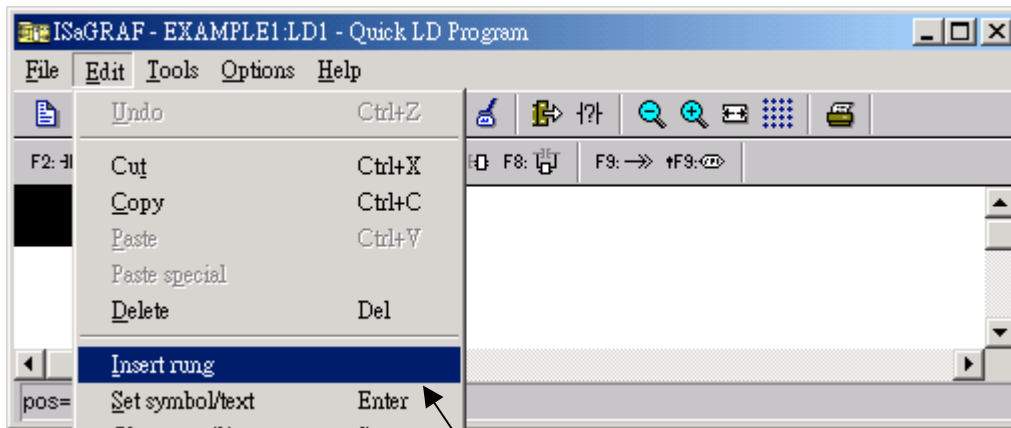


Now we have one program inside this project. Please double click on the "LD1" to get into it.

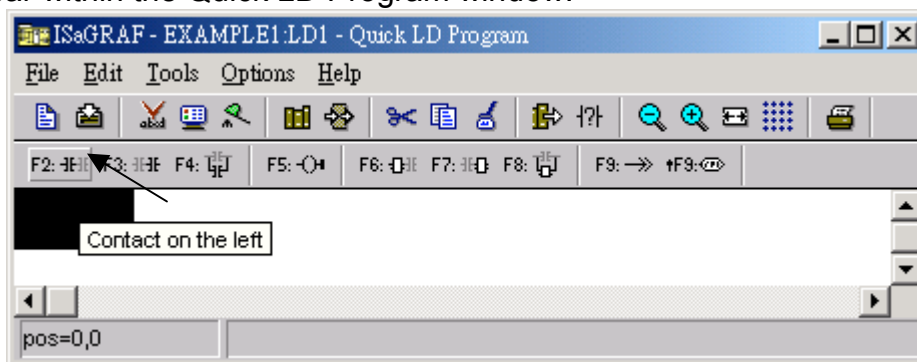


4.1.7: Edit The "LD1" Program

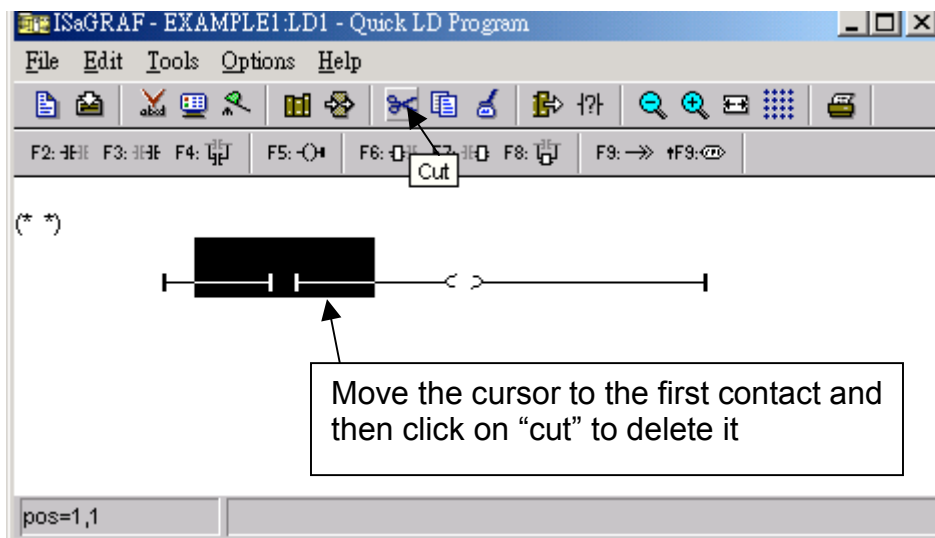
When you double click on the "LD1" name the "Quick LD Program" window will appear. To start programming our LD program, click on "Edit" from the main menu bar, then click on "Insert Rung". "Insert Rung" means to insert a basic LD rung just above the current position.



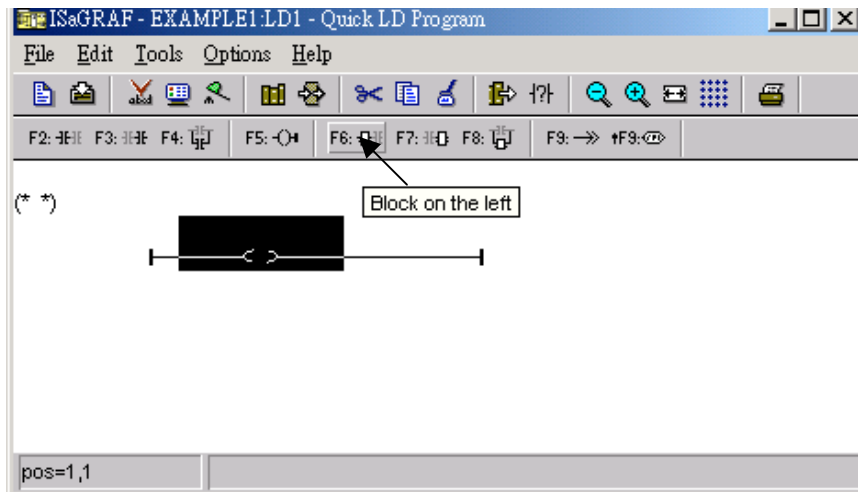
Or, you may just simply click on the "F2 (Contact On The Left)" icon, and the following will appear within the Quick LD Program window.



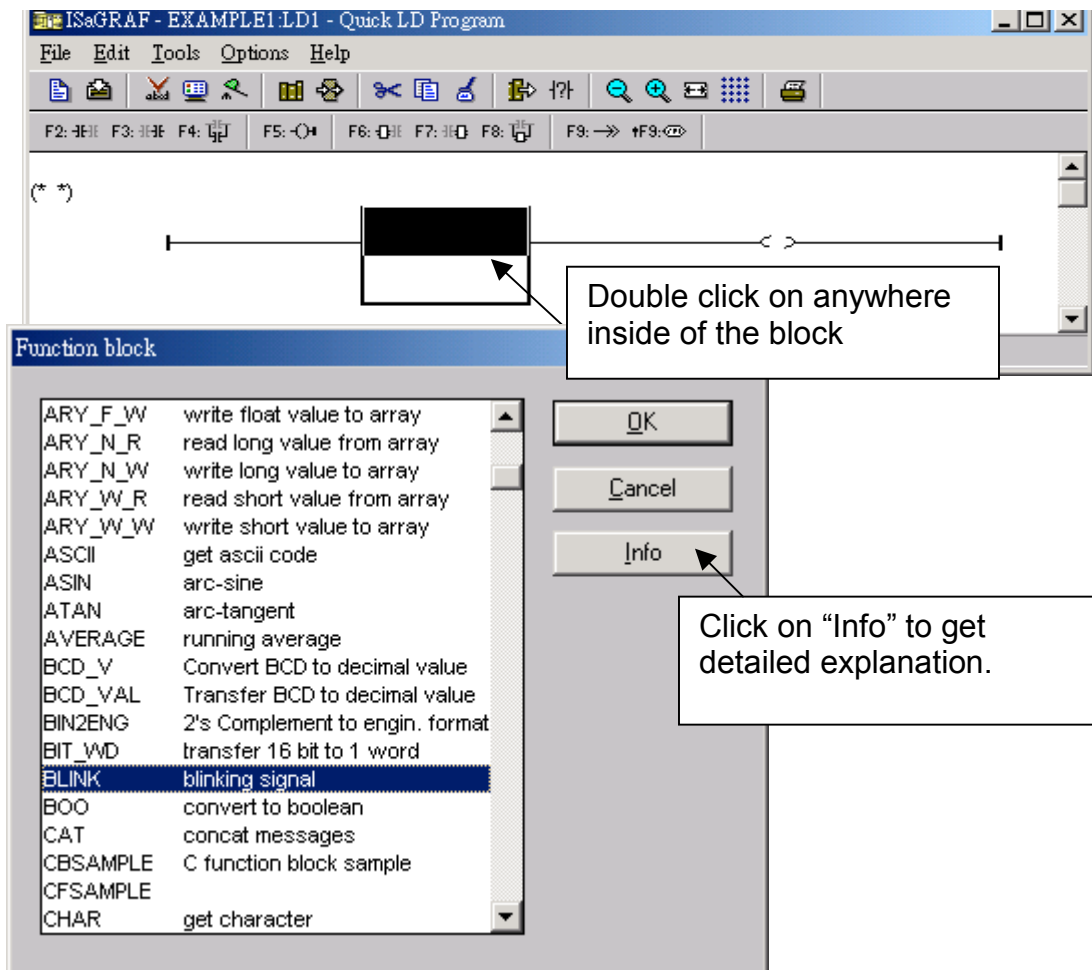
We are going to write the first line of the LD1 program. Move the cursor to the first "contact" and then click on "cut" to delete it.



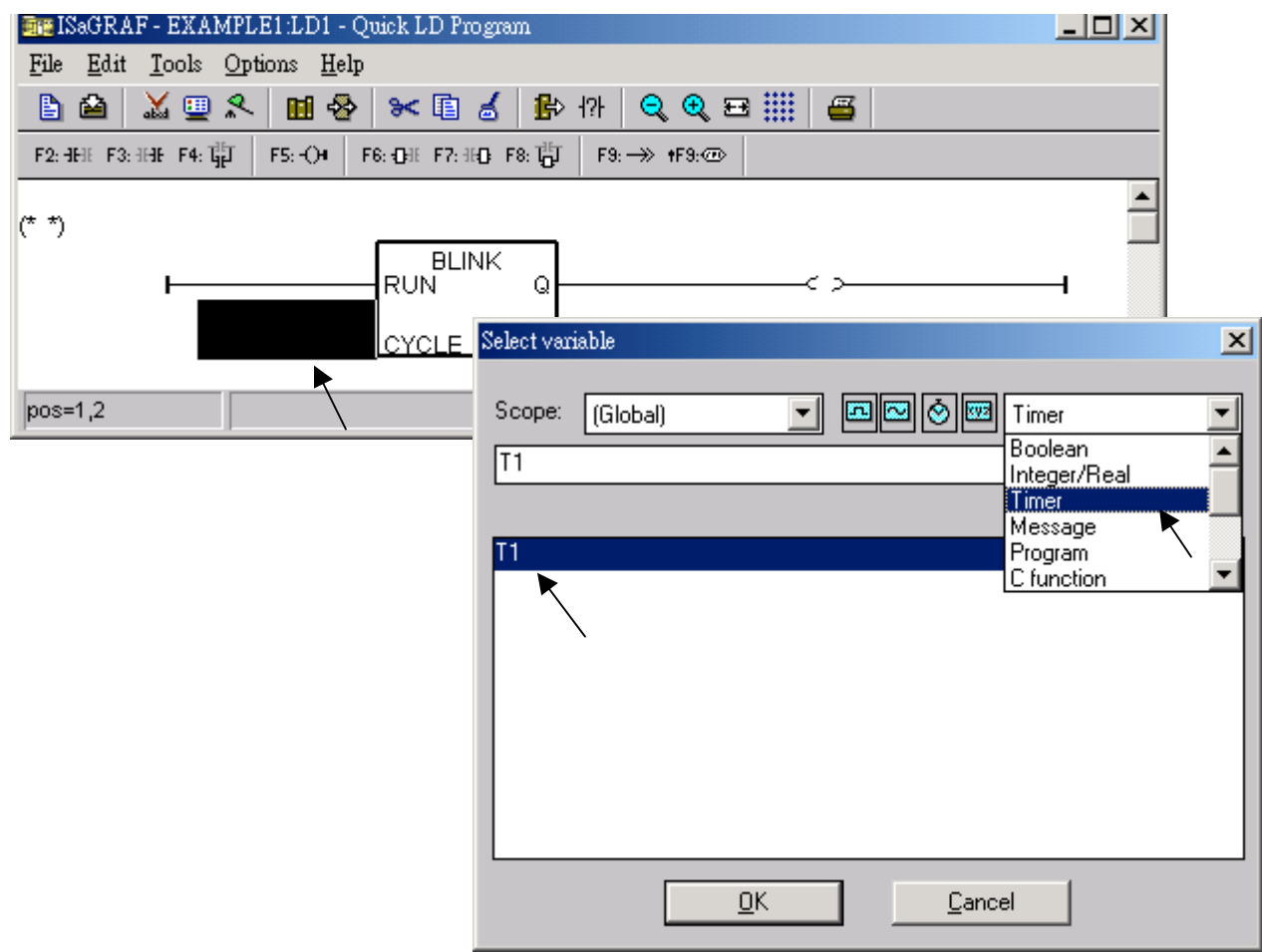
Click on the "F6 (Block on the left)" icon and you will create a block on the left of the "coil".



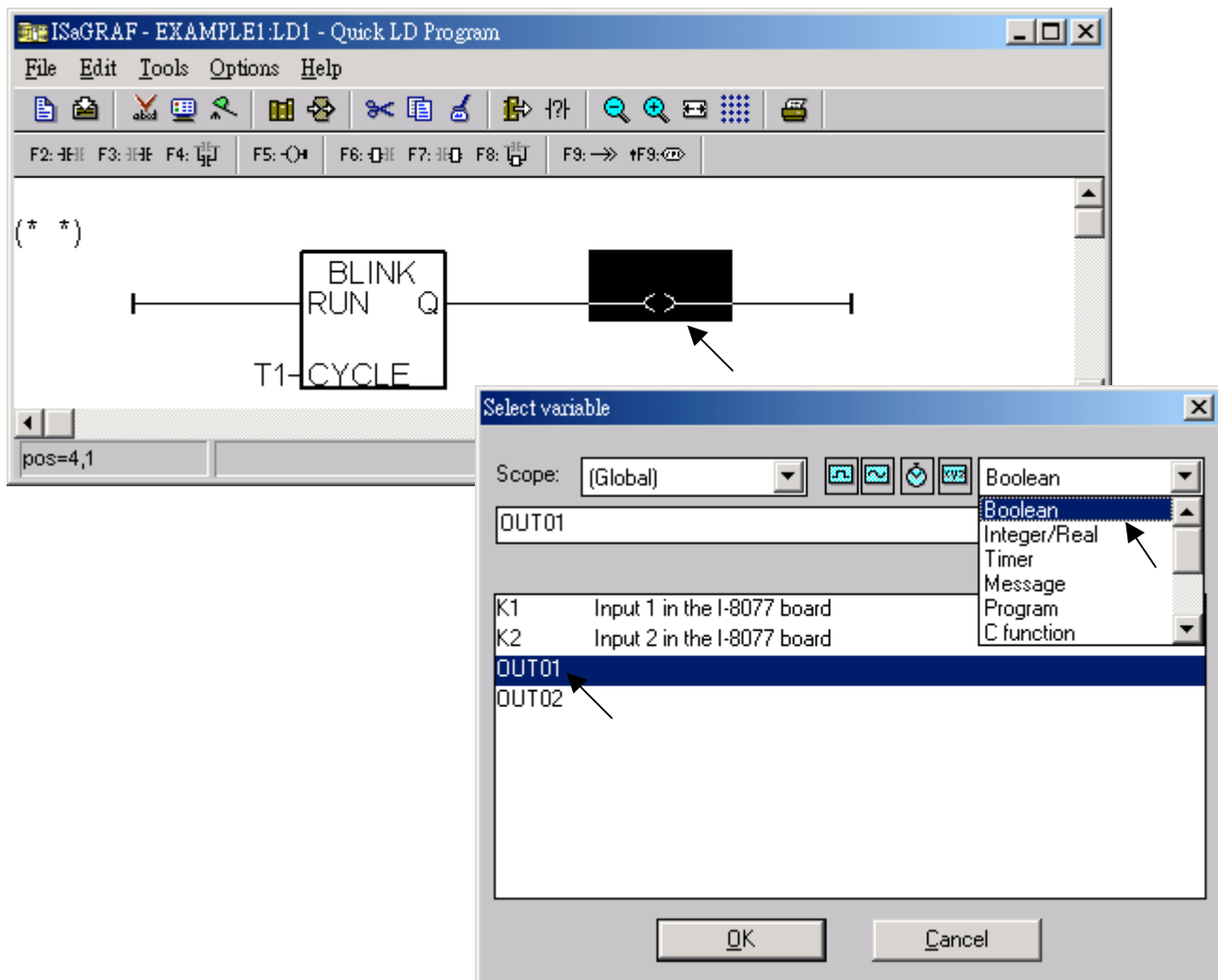
Now we are going to assign the associated variable & constant to each item. Double click anywhere inside of the block and the "Function Block" assignment window appears. Select the "BLINK" type function block . To learn how the "BLINK" function operates you can click on the "Info" button for a detailed explanation of its functionality



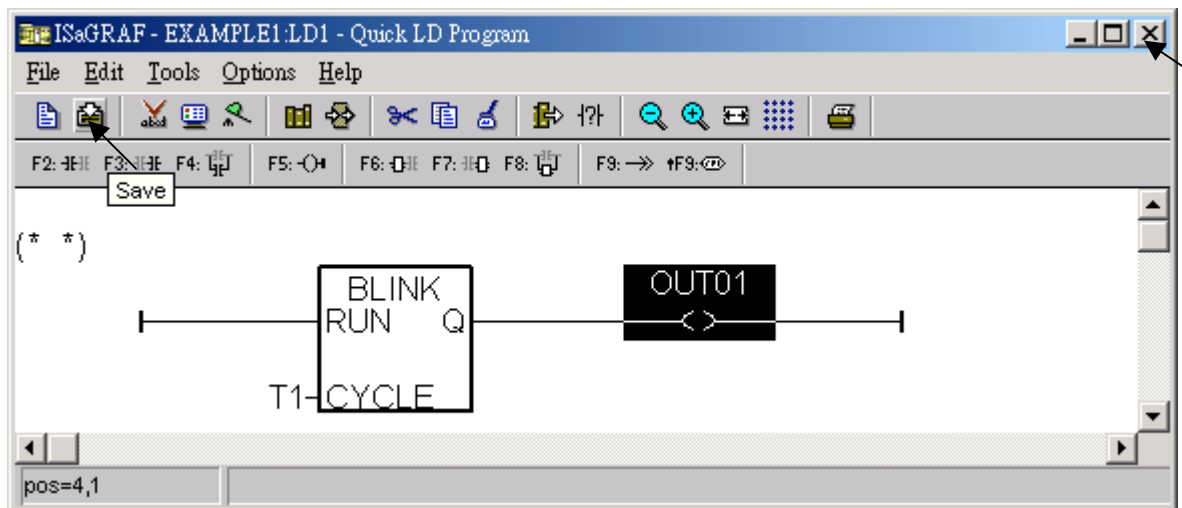
Now move your cursor to the left of the parameter “CYCLE” of the “BLINK” block. Double click on it, select “Timer” and then double click on variable name - “T1”



Move your cursor to the “coil”. Double click on it, select “Boolean” and then double click on variable name – “OUT01”.

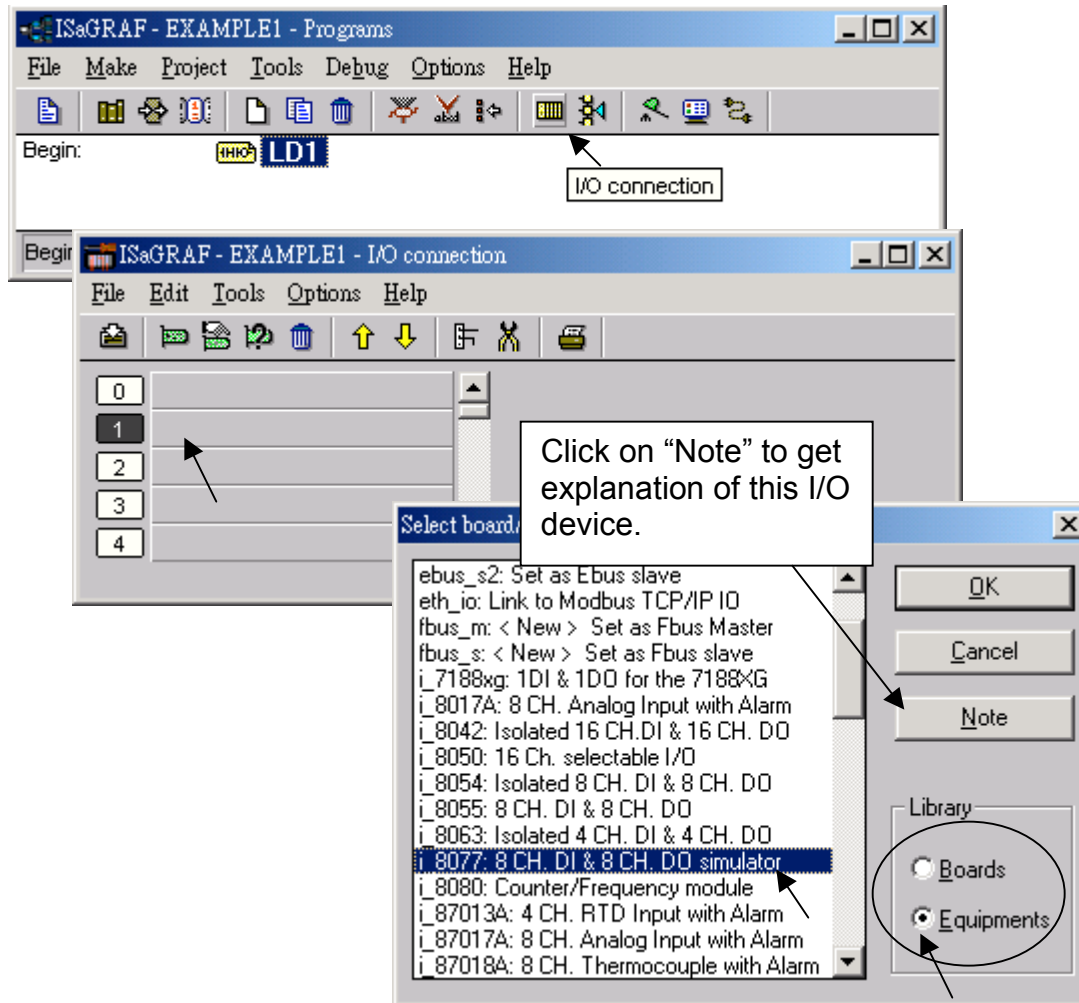


Now we have finished our Ladder code, click on “Save” and then click on “X” to exit.

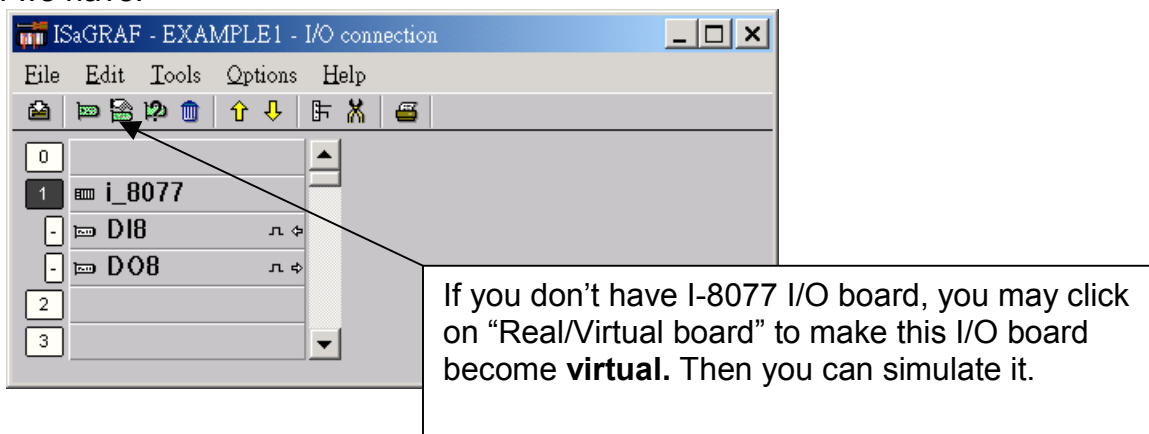


4.1.8: Connecting The I/O

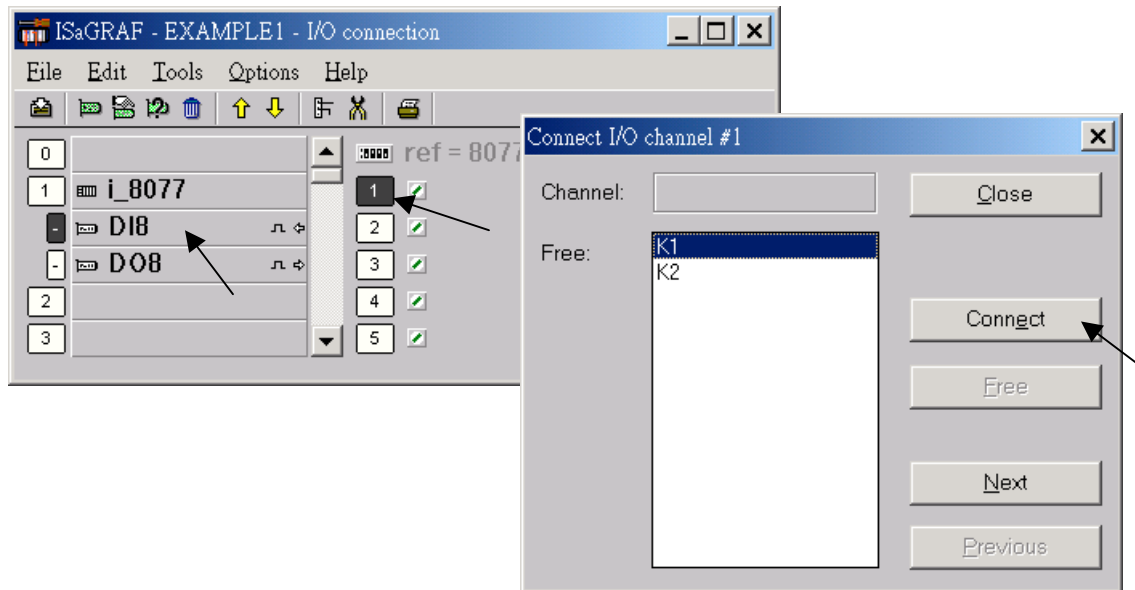
We have defined variables name of “OUT01” , “OUT02” as “output” attribution, while “K1” & “K2” as “input” attribution in step 4.1.4. These “input” & “output” variables should be map to physical I/O in the controller before they can work. To do that, click on “I/O connection” to get into the I/O connection window. Double click on the No. 1 slot (Please make sure your I-8077 I/O board is plug in slot 1) & then check on the “Equipments” & double click on the “I_8077: 8 CH. DI & 8 CH. DO simulator”.



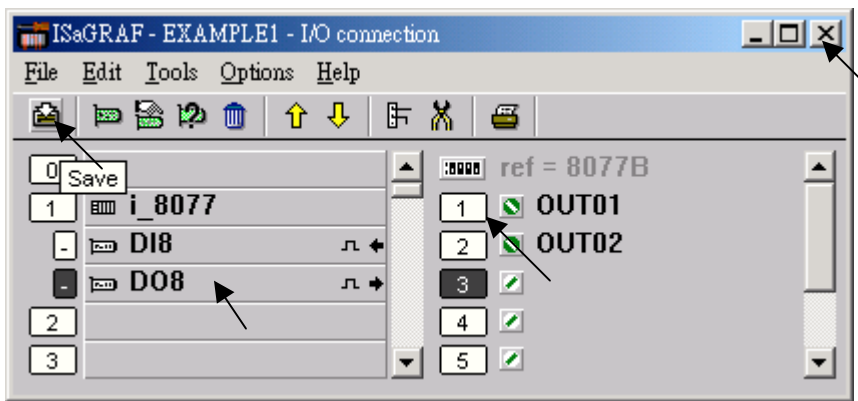
Then we have.



To map input variables “K1” & “K2” to the input channel No. 1 & 2 of the “I-7077”, double click on the channel 1 and then click on “Connect” .Then click on “Connect” again to connect channel 2.



By the same way, please connect “OUT01” , “OUTPUT02” to output channel 1 to 2. Then we have below window. Click on “Save” and then exit.



IMPORTANT NOTICE:

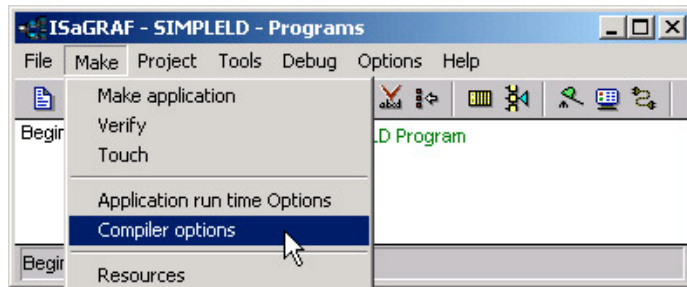
1. I/O Slots 1 through 7 are reserved for REAL I/O boards that will be used in the W-8xx7. You can use slot No. 8 and above for additional functionality.

2. All of the variables with “Input” and “Output” attribute MUST be connected through the I/O connection as described above for any program to be successfully compiled. Only the Input and Output attributed variables will appear in the "I/O Connections" window. In this example we have only 2 boolean output variables - OUT01, OUT02 and 2 boolean input variables – K1 & K2.

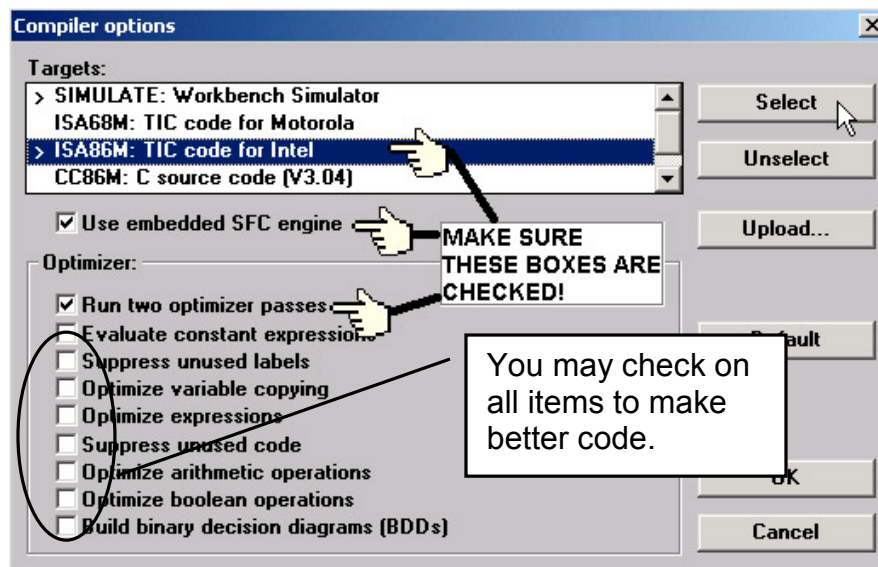
4.2: Compiling & Simulating The Example Project

For ANY AND EVERY ISaGRAF program to work properly with any of the I-7188EG, 7188XG, I-8417/8817/8437/8837 , Wincon-8037/8337/8737 controller systems, it is the responsibility of the programmer to properly select the correct "Compiler Options". You **MUST** select the "ISA86M: TIC Code For Intel" option as described below.

To begin the compilation process, first click on the "MAKE" option from the main menu bar, and then click on "Compiler Options" as shown below.

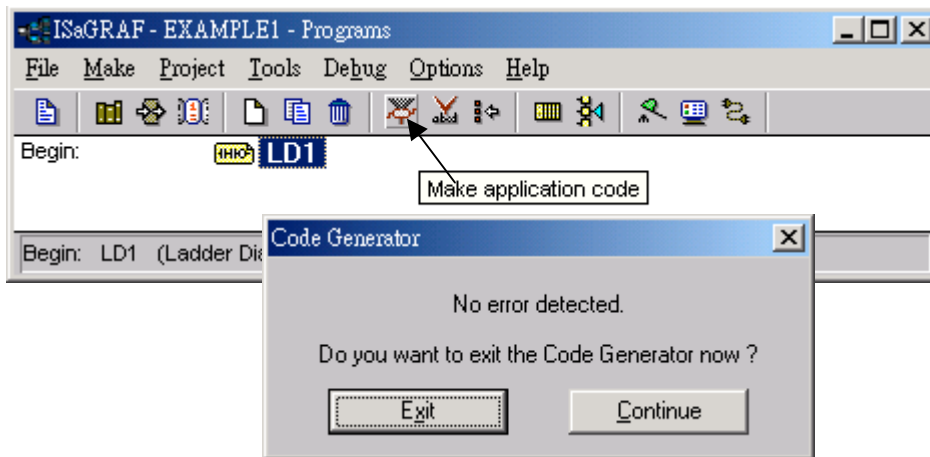


The "Compiler Options" window will now appear. Make sure to select the options as shown below then press the "OK" button to complete the compiler option selections.



TIME TO COMPILE THE PROJECT!

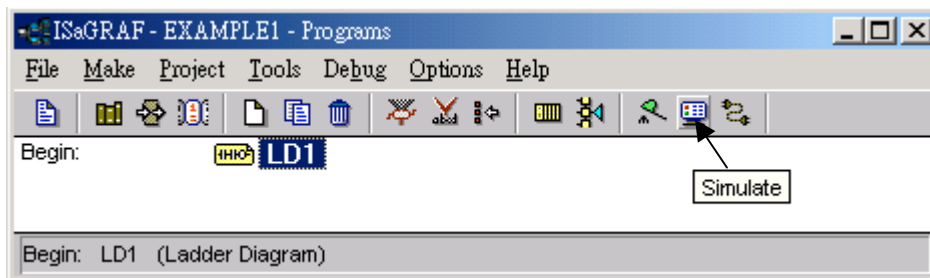
Now that you have selected the proper compiler options, click on the "Make Application Code" icon to compile the example project. If there are no compiler errors detected during the compilation process, CONGRATULATIONS, you have successfully created our example program.



If errors are detected during the compilation process, just click on the "CONTINUE" button to review the error messages. Return to the Project Editor and correct the errors as outlined in the error message window.

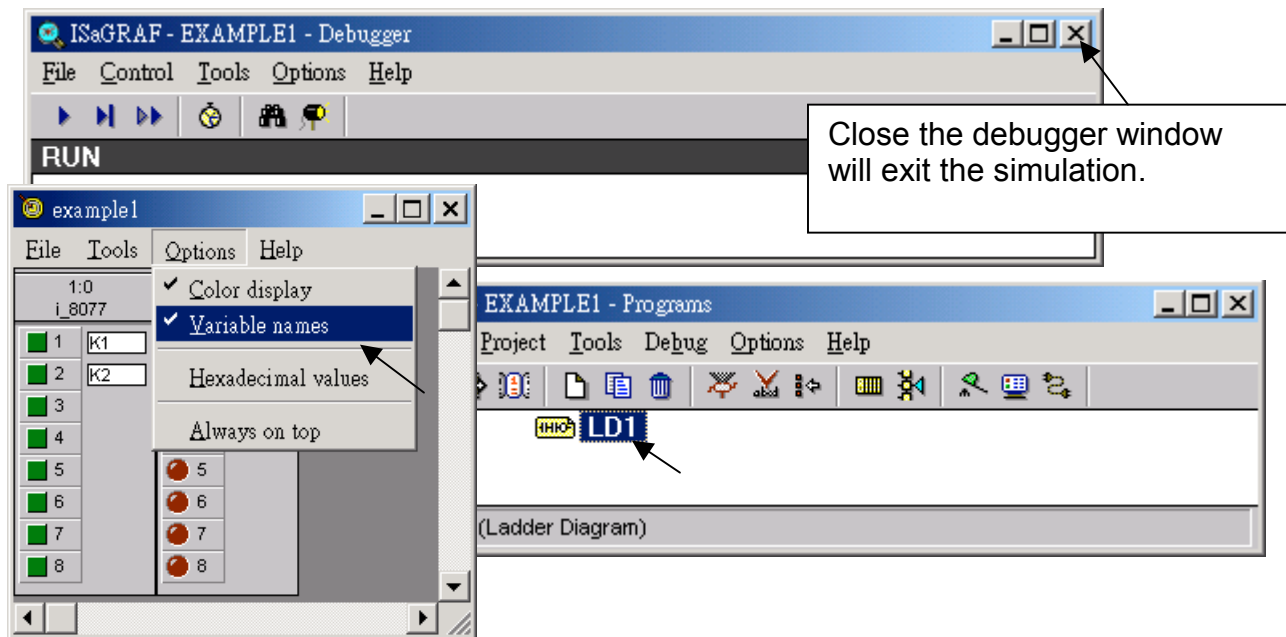
TIME TO SIMULATE THE PROJECT!

If the compilation is Ok, you may simulate the project on the PC to see how the program works without the controller. To do that, click on the "Simulate" icon.



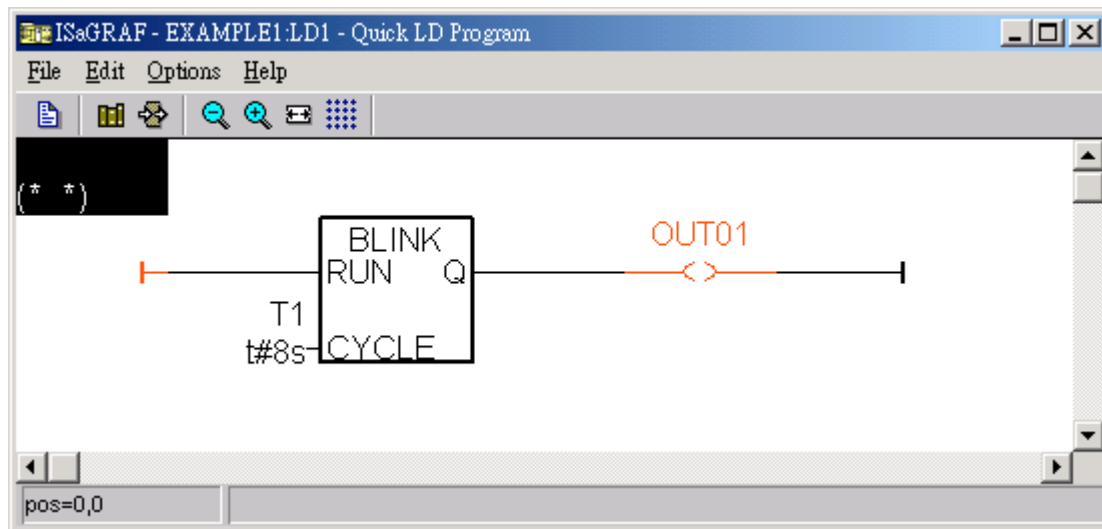
When you click on the "Simulate" icon three windows will appear. The windows are the "ISaGRAF Debugger", the "ISaGRAF Debug Programs", and the "I/O Simulator" windows. If the I/O variable names you have created DO NOT appear in the I/O simulator window, just click on the "Options" and "Variable Names" selection and the variable names you have created will now appear next to each of the I/O's in the simulator window.

In the "ISaGRAF Debug Program" window, double click on the "LD1" where the cursor below is positioned. This will open up the ISaGRAF Quick LD Program window and you can see the LD program you have created.



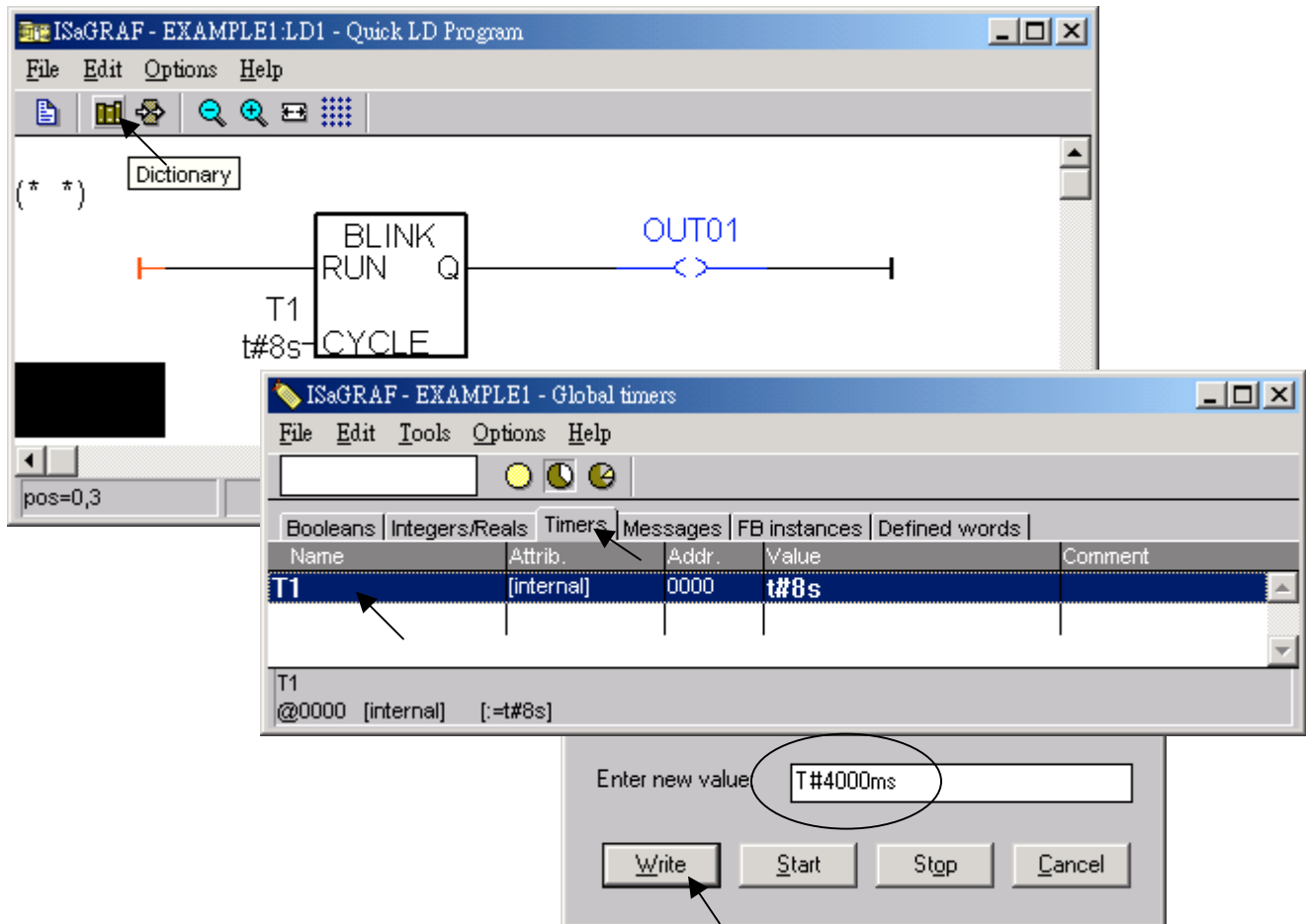
Running The Simulation Program

When you double click on "LD1" in the "ISaGRAF Debug Programs" window, the follow window should appear.

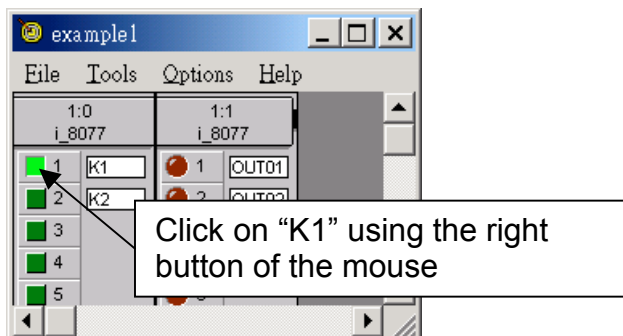


You can see outputs "OUT01" will blink in the period of 8 seconds.

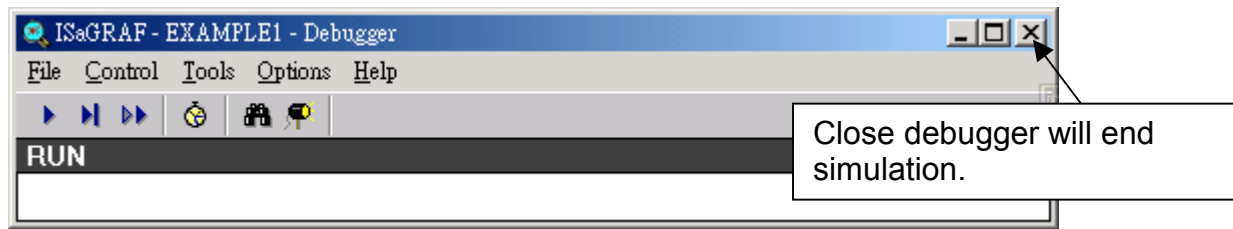
You can adjust the "T1" variable while the program is running. To accomplish this, click on the "Dictionary" icon which will open the "ISaGRAF Global Variables" window as shown in the first two pictures below. Click on "Timer" tab and then double click on "T1" to change the timer value to "T#4000ms" (this means 4000 ms). Then click on "Write".



Now we are going to simulate the "K1" & "K2" input. Click on "K1" using the right button of the mouse.



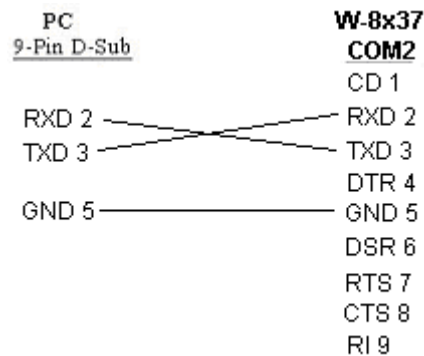
To exit simulation, please close the debugger window.



4.3: Download & Debug The Example Project

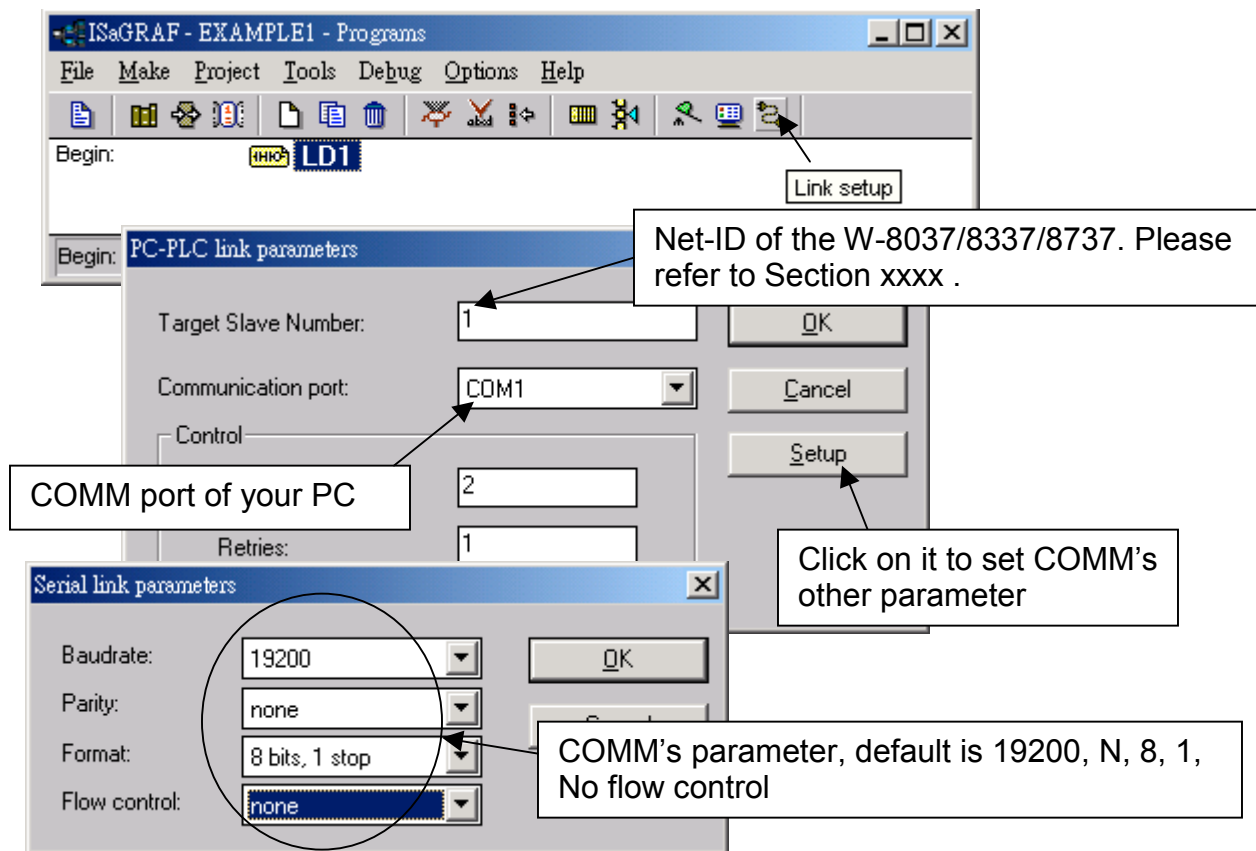
We have 2 ways to download the project to the Wincon. One is using Ethernet cable, the other one is using RS232 cable. Here will show you the RS232 way. **(Please refer to section 3.2.4.1 if you would like to download the project via ethernet)**

To begin this process, please install the hardware as below. The RS232 cable wiring should be 2, 3 cross as below figure. **(Please make sure the “Modbus RTU Slave Port” is set as COM2 (refer to Appendix A.2, or it can only be download via ethernet)**



This section lists how to download the ISaGRAF program via RS232 cable. However user may also use Ethernet cable to download program to the Wincon-8xx7 (please refer to section 3.2.4.1)

Click on the "Link Setup" icon in the "ISaGRAF Programs" window. When you click on the "Link Setup" icon, the following window will appear. Please set the proper value.

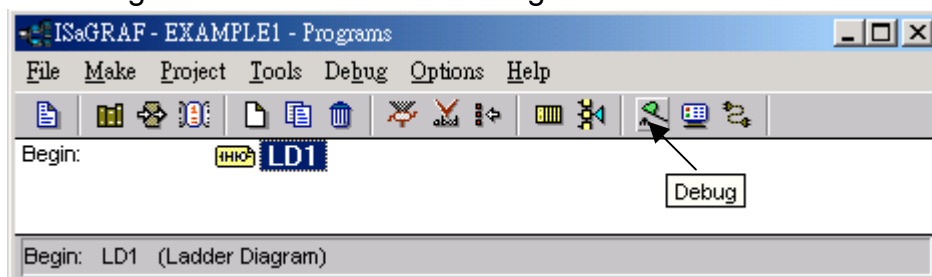


The RS232 communication parameters for the target W-8037/8337/8737 controller MUST be set to the same serial communication parameters for the development PC. For W-8xx7 controllers (serial port communications), the default parameters for COM2 (RS232) port are:

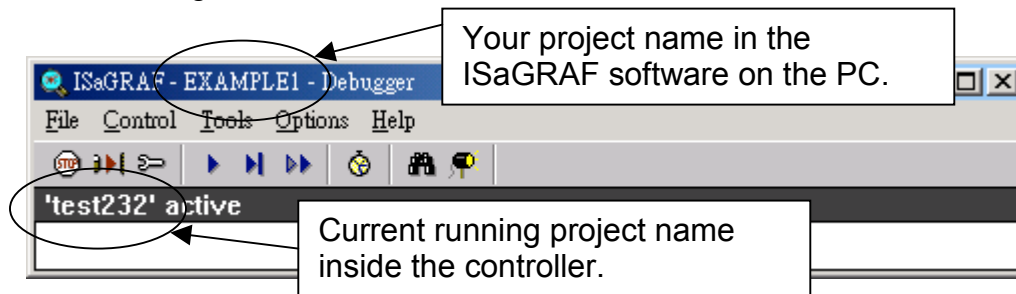
Baudrate:	19200
Parity:	none
Format:	8 bits, 1 stop
Flow control:	none

DOWNLOADING THE EXAMPLE PROJECT

Before you can download the project to the controller, you must first verify that your PC and the controller system are communicating with each other. To verify proper communication, click on the "Debug" icon in the "ISaGRAF Programs" window as shown below.



If the development PC and the W-8037/8337/8737 controller system are communicating properly with each other, the following window displayed below will appear (or if a program is already loaded in the controller system, the name of the project will be displayed with the word "active" following it).

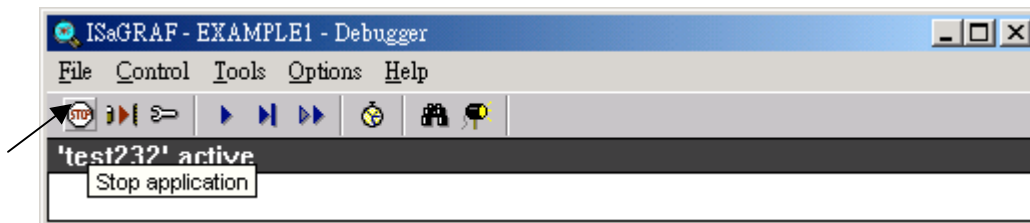


If the message in the "ISaGRAF Debugger" says "Disconnected", it means that the development PC and the controller system have not established communications with each other.

The most common causes for this problem is either the serial port cable not being properly configured, or the development PC's serial port communications DO NOT match that of the W-8xx7 controller system.

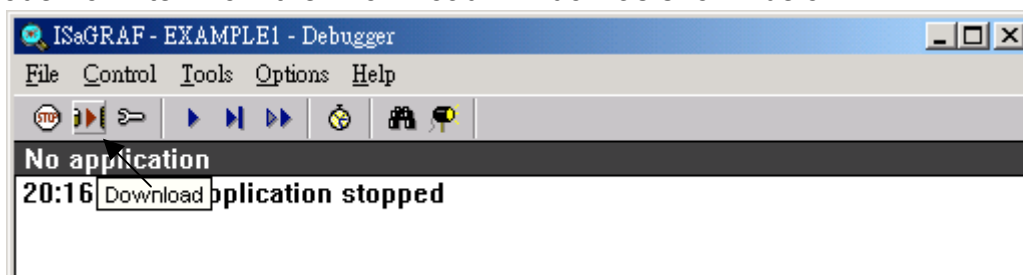
You may have to either change the serial port communication settings for the development PC (which may require changing a BIOS setting) or change the "Serial Link Parameters" in the ISaGRAF program.

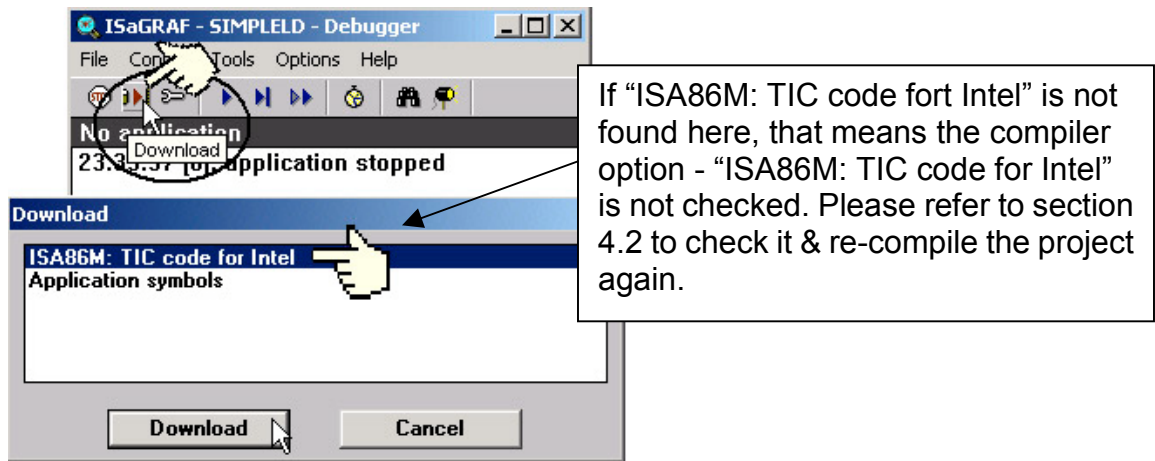
If there is a project already loaded in the controller system you will need to stop that project before you can download the example project. Click on the "STOP" icon as illustrated above to halt any applications that may be running.



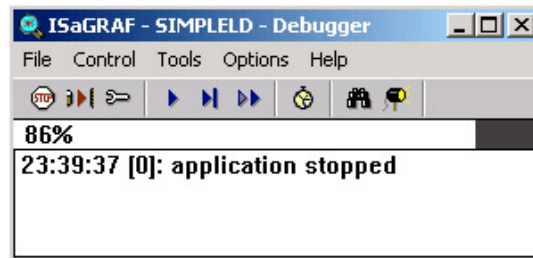
STARTING THE DOWNLOADING PROCESS

From the "ISaGRAF Debugger" window click on the "Download" icon, then click on "ISA86M: TIC Code For Intel" from the "Download" window as shown below.

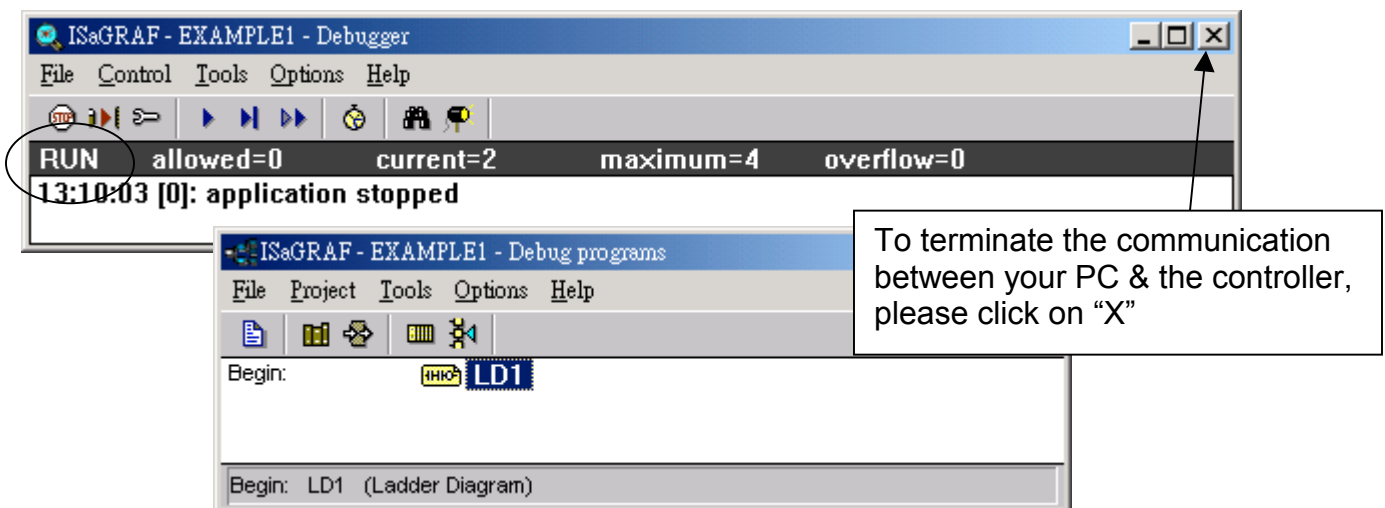




The example project will now start downloading to the W-8xx7 controller system. A progress bar will appear in the "ISaGRAF Debugger" window showing the project downloading progress.

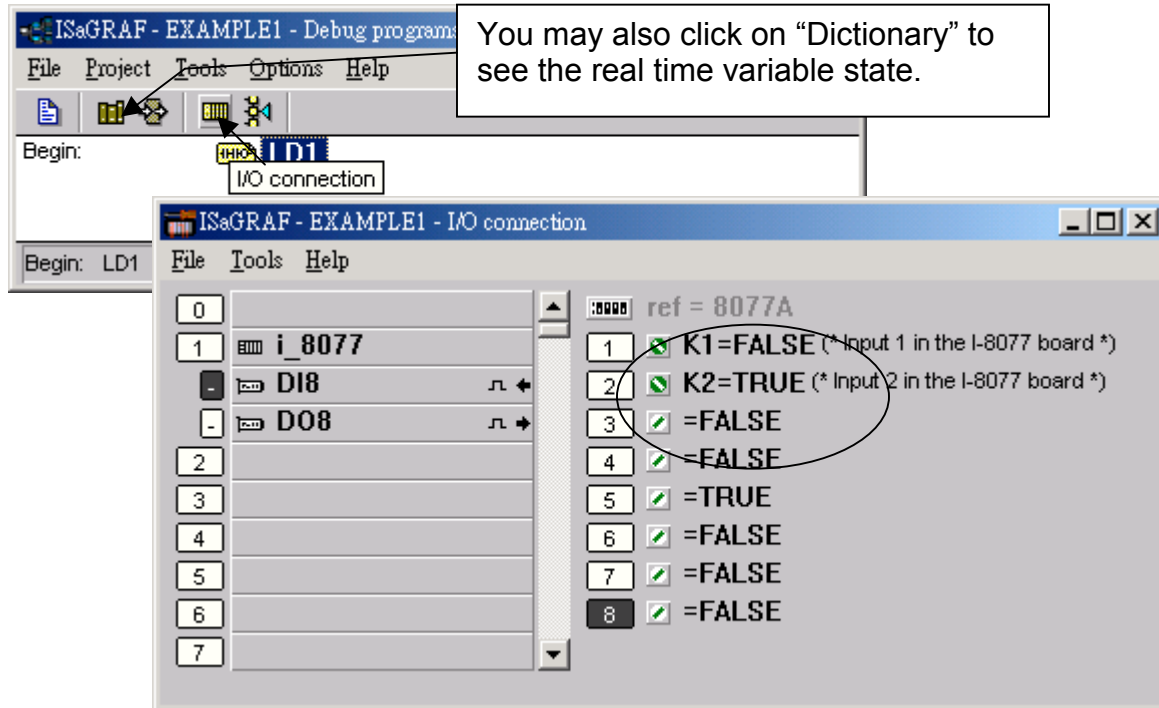


When the example project has successfully completed the downloading process to the W-8xx7 controller system the following two windows will appear.

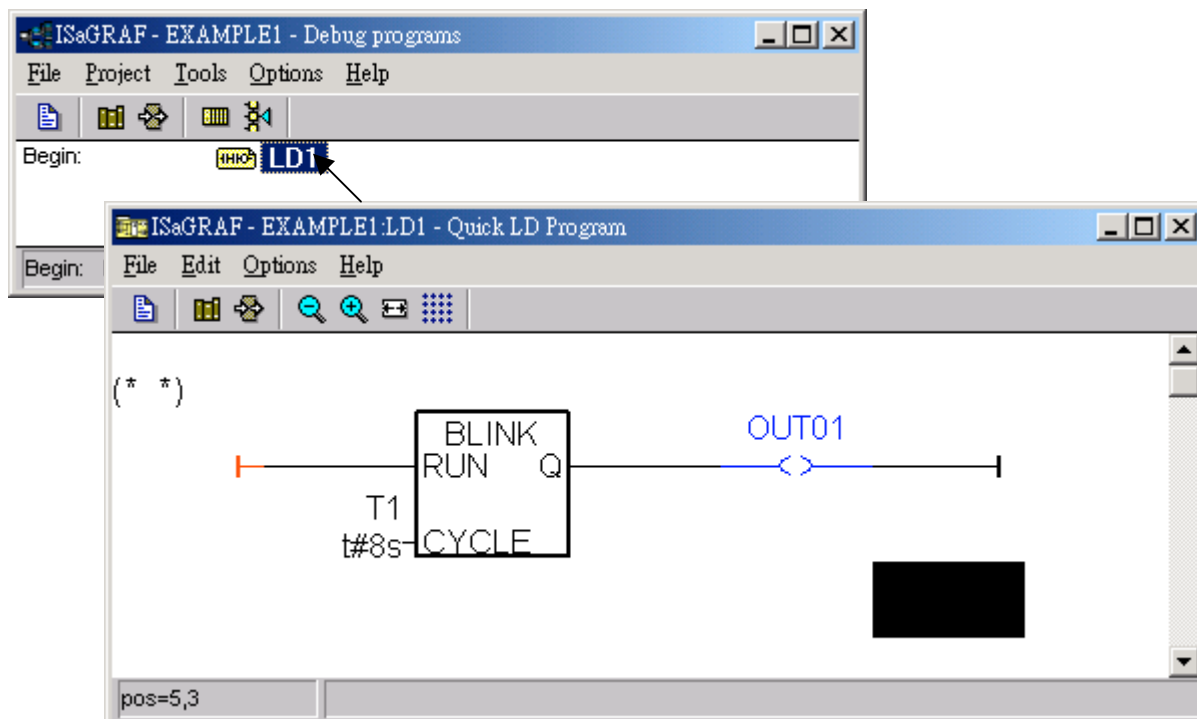


RUNNING THE EXAMPLE LD PROGRAM

You can observe the real time I/O status from several ISaGRAF windows while you are running the example project. One of the windows is the "I/O Connections" window, which shows each of the inputs and outputs as assigned. Click on the "I/O Connections" icon in the ISaGRAF Debugger window to open the "I/O Connections" screen. You may switch ON/OFF the D/I on the front panel of the I-8077 I/O board to see what happens about "K1" & "K2"



Another VERY helpful window you can open is the "Quick LD Program" window. From this window you can observe the LD program being executed in real time.



4.4: Design The Web Page

After finishing the ISaGRAF project & download it to the Wincon-8x37, we are going to design the Web Page for this ISaGRAF project.

If you haven't practiced "Setting Up A Web HMI Demo" listed in the Chapter 3, it's better to do it once to get familiar with it.

We will use "**Microsoft Office FrontPage 2003**" (or higher version) to build web pages in this manual. User may choose your prefer web page editor to do the same thing.

You may refer to the finished web pages of this example in the W-8x37 CD-ROM at design time. However it is better to do it one time by yourself to get more understanding.

`\napdos\isagraf\wincon\WebHMI_Demo\example1`

4.4.1: Step 1 – Copy The Sample Web HMI pages

These is a sample Web HMI pages in the Wincon-8x37 CD-ROM:

`\napdos\isagraf\wincon\WebHMI_Demo\sample`

Please copy this "sample" folder to your hard drive and rename it, for example, rename it as "**example1**".

The basic Web HMI files includes 2 folders and 3 DLL files and 4 htm files as below.

<code>./img/</code>	(default image files - *.jpg , *.bmp , *.gif)
<code>./msg/</code>	(default message files – wincon.js & xxerror.htm)
<code>whmi_filter.dll</code>	(three DLL files)
<code>login.dll</code>	
<code>main.dll</code>	
<code>index.htm</code>	(first default page)
<code>login.htm</code>	(the Web HMI welcome page)
<code>menu.htm</code>	(the page-menu page, normally on the left on the Internet Explorer)
<code>main.htm</code>	(first page when successfully login)

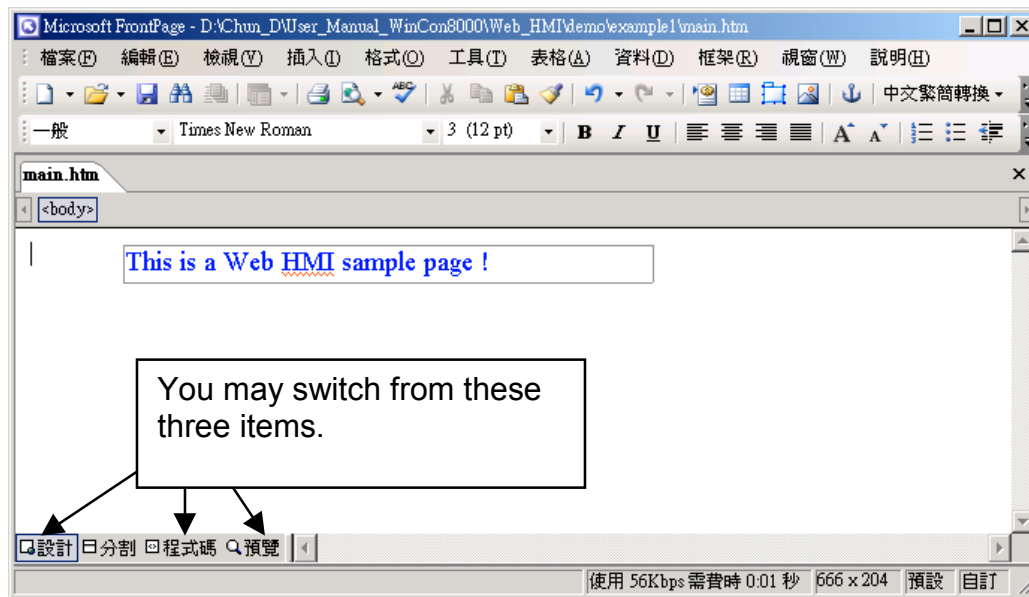
User may put his own image files into the folder named as "user_img". And put user-defined javascript file or css file into the folder named as "user_msg". Other folder name is not acceptable by the Wincon Web HMI.

The "index.htm" file is the default entry page of the web server. User should not modify it. The "index.htm" re-directs to the "login.htm" file in 1 to 2 second when someone visits the Wincon-8x37 via the Internet Explorer.

User may modify the "login.htm" , "menu.htm" & "main.htm" to fit his own need. We will only modify the "main.htm" in this example.

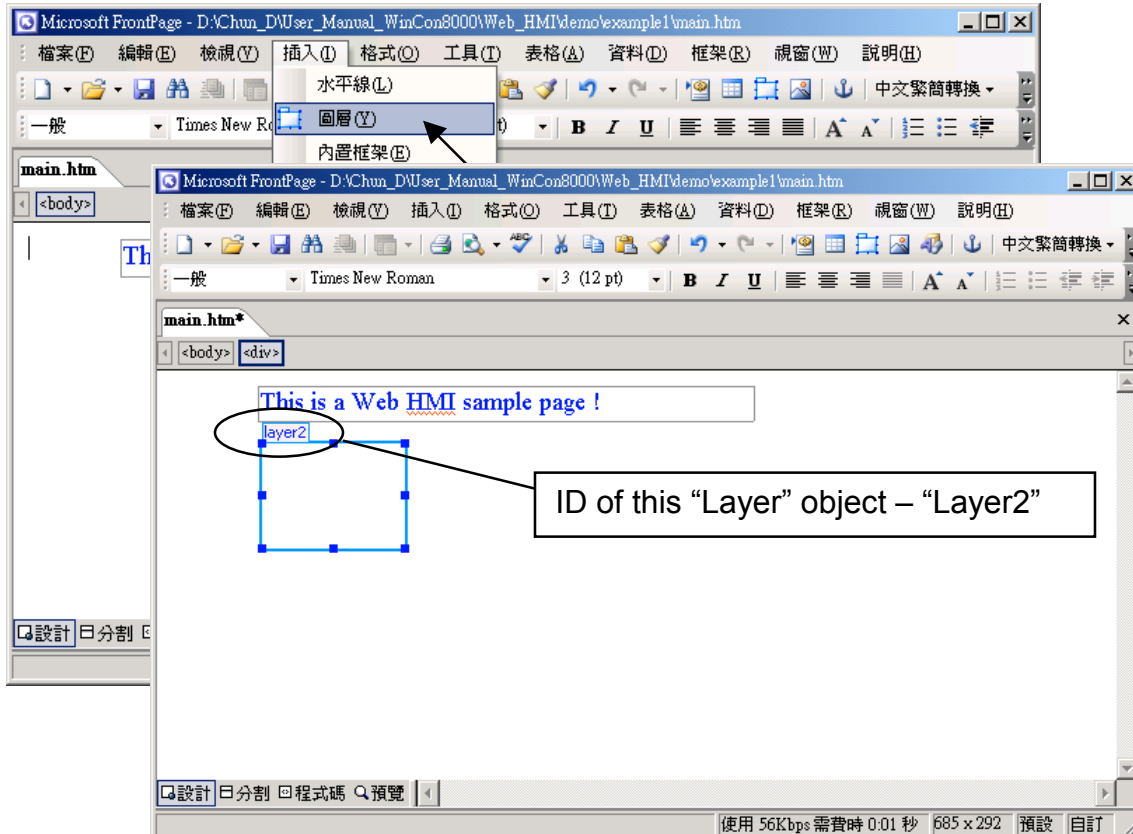
4.4.2: Step 2 – Building The Main.htm

Please run the Microsoft Office FrontPage 2003 (or higher version) and open the “main.htm”

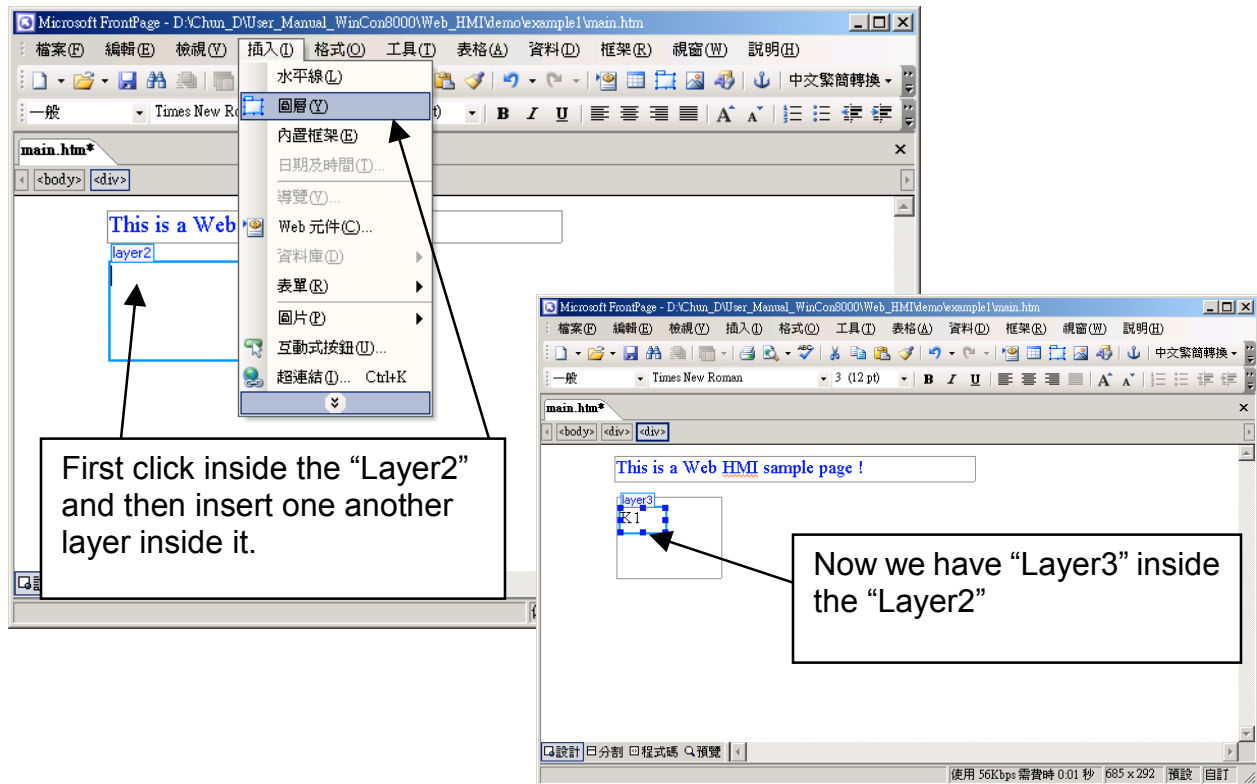


Please switch the window to design the page.

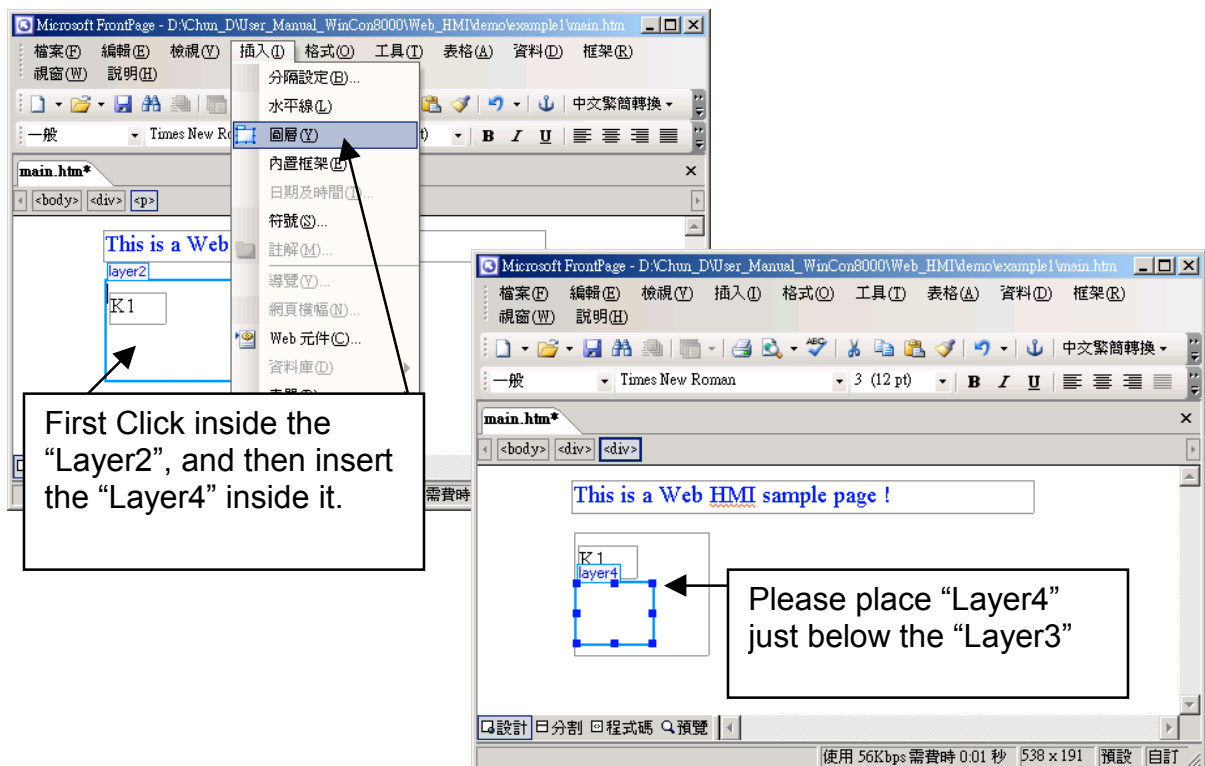
Please insert a layout object – “Layer” as below.



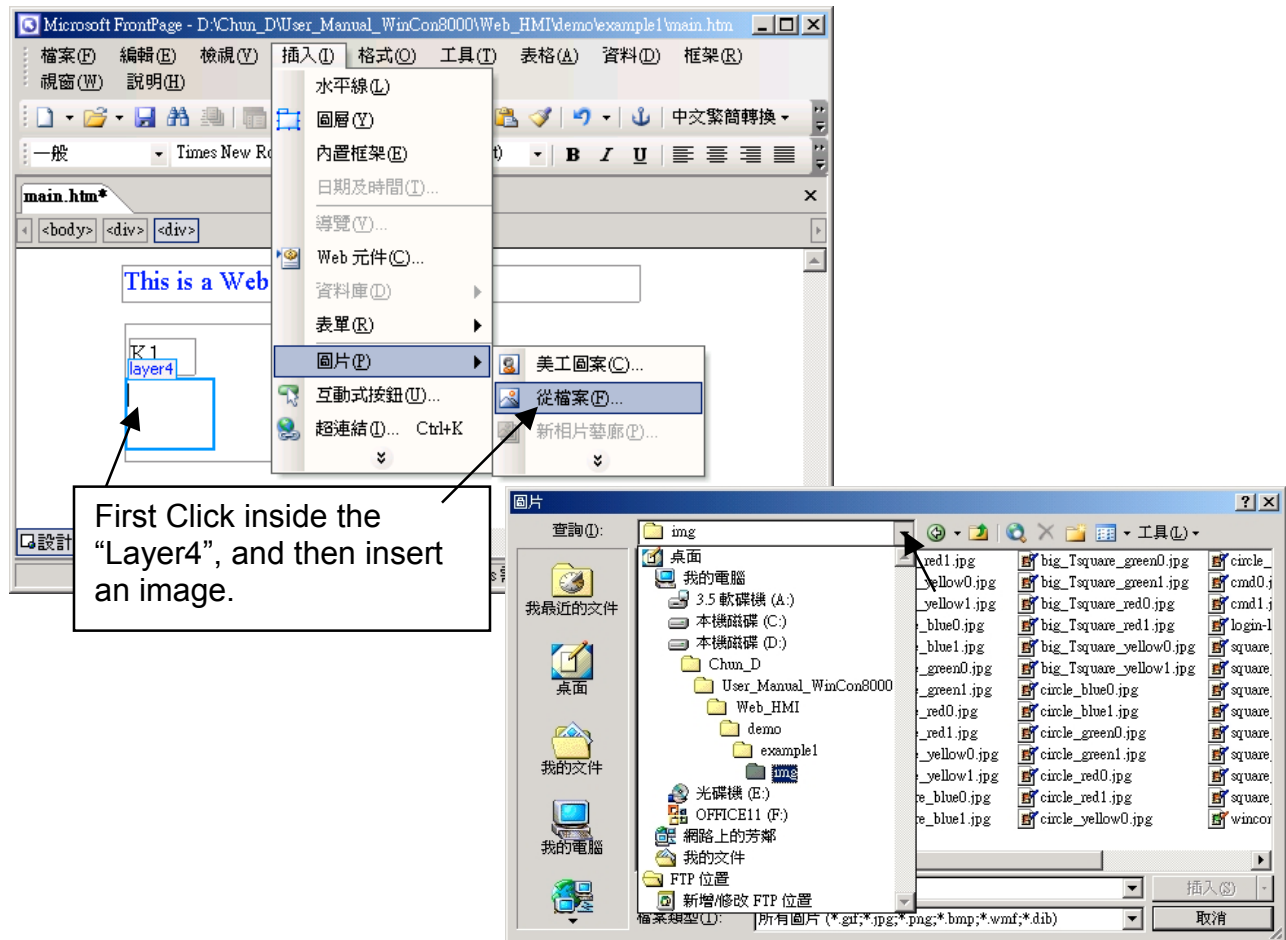
Click inside this “Layer” and then insert one another layer inside it as below. Please enter “K1” into the new created “Layer”.



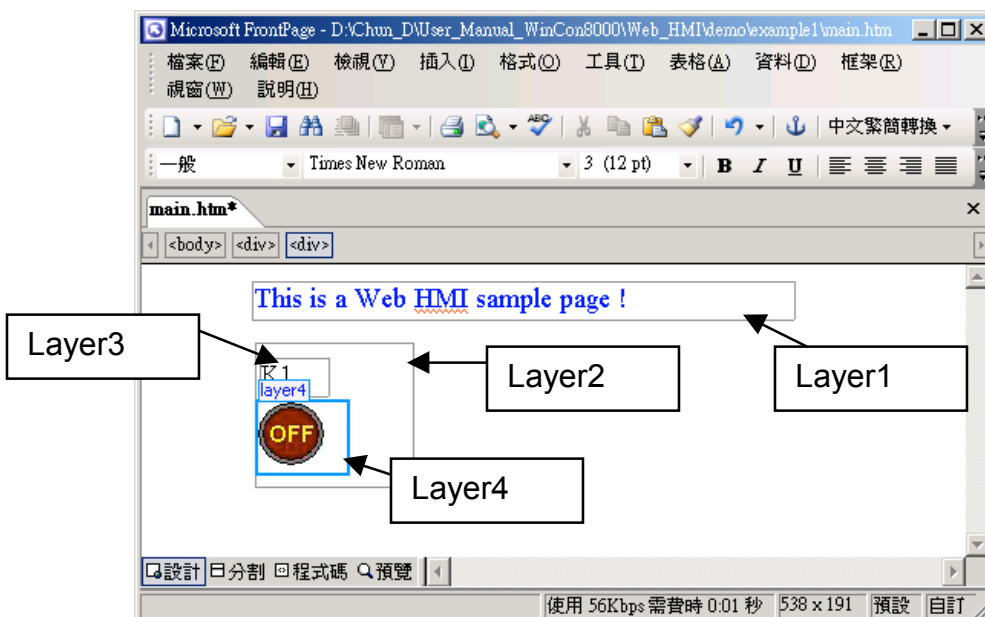
Follow the same former steps to insert one another “Layer” to be in just below the “Layer3” as below.



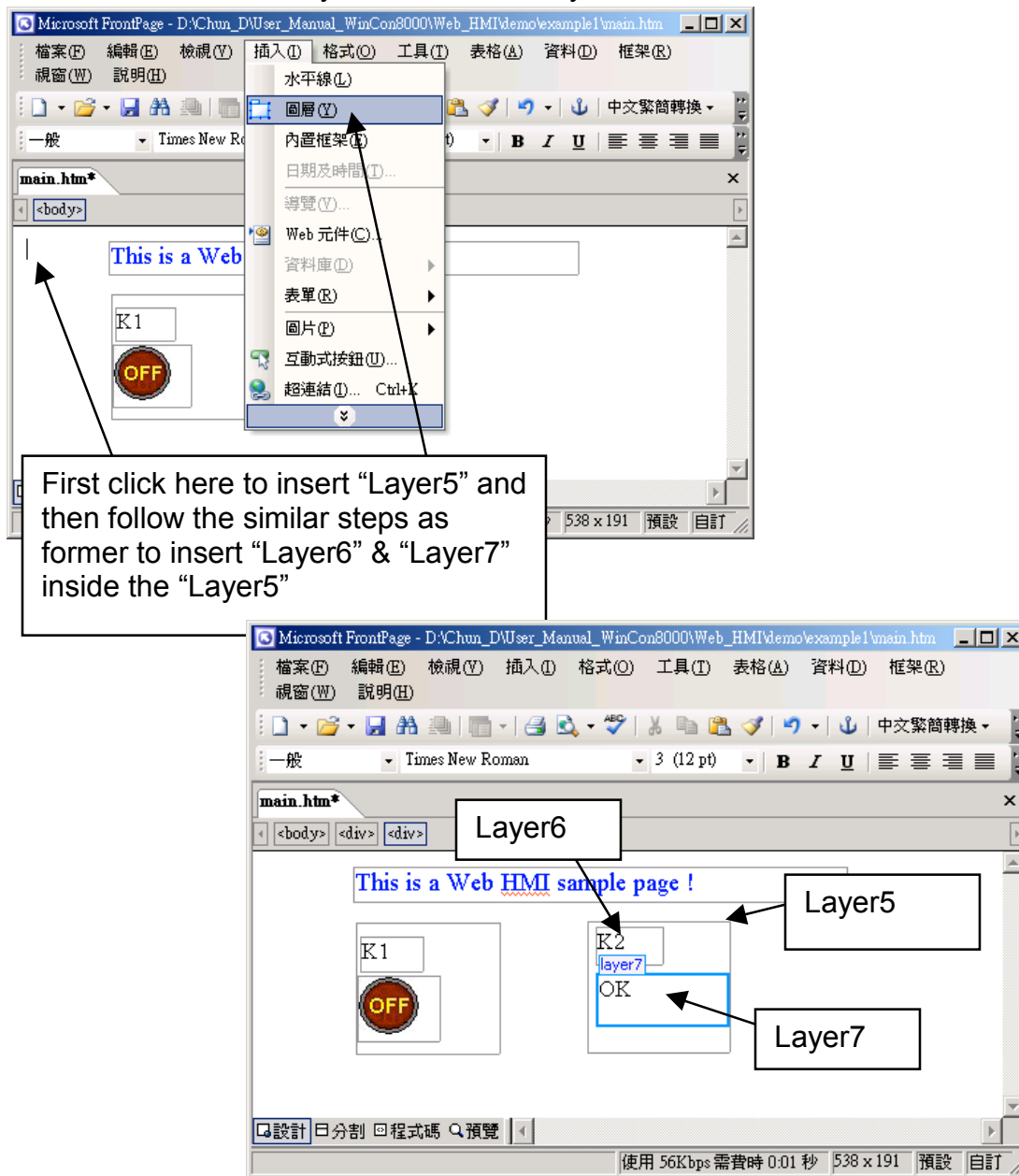
Inside the “Layer4”, we are going to insert one image file to it as below. The image file name is “./img/big_Tcircle_red0.jpg”. Please browse to the correct folder in your hard driver. Here we use “example1/img/” in this example.



You will see a window as below.

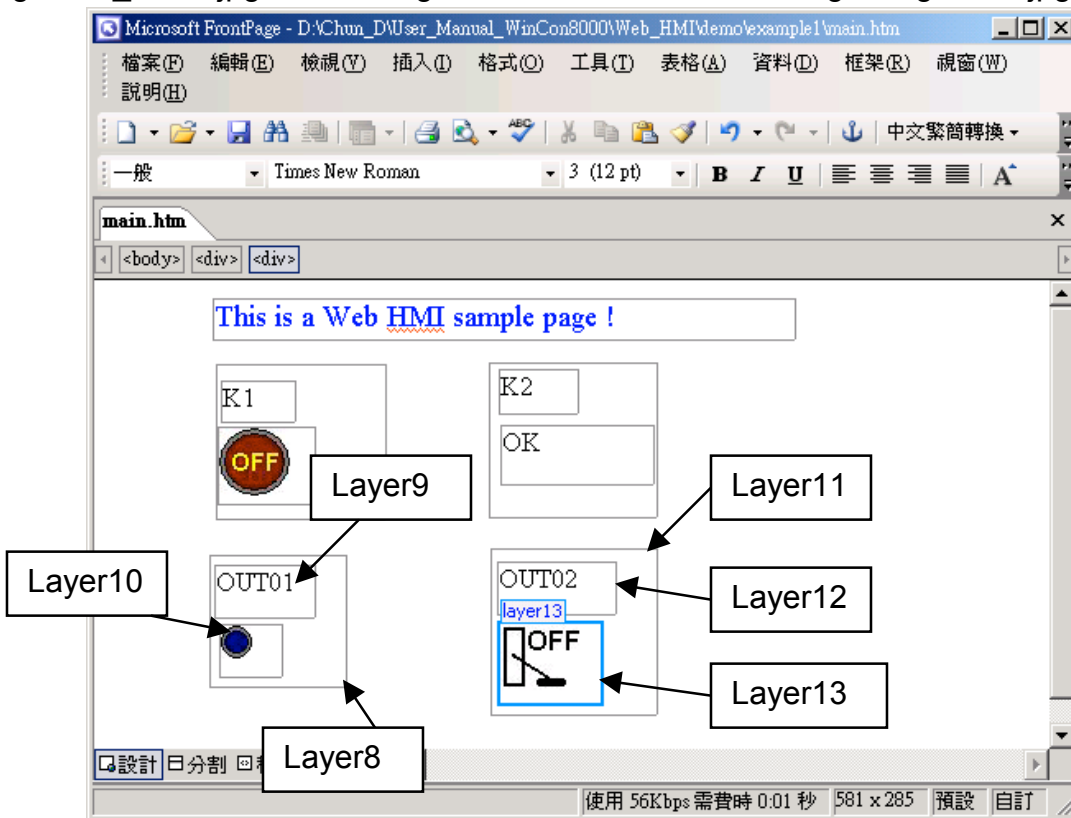


Please follow the similar steps to insert one another “Layer5” and one “Layer6” with a “K2” symbol inside it, and also a “Layer7” with a “OK” symbol inside it as below.



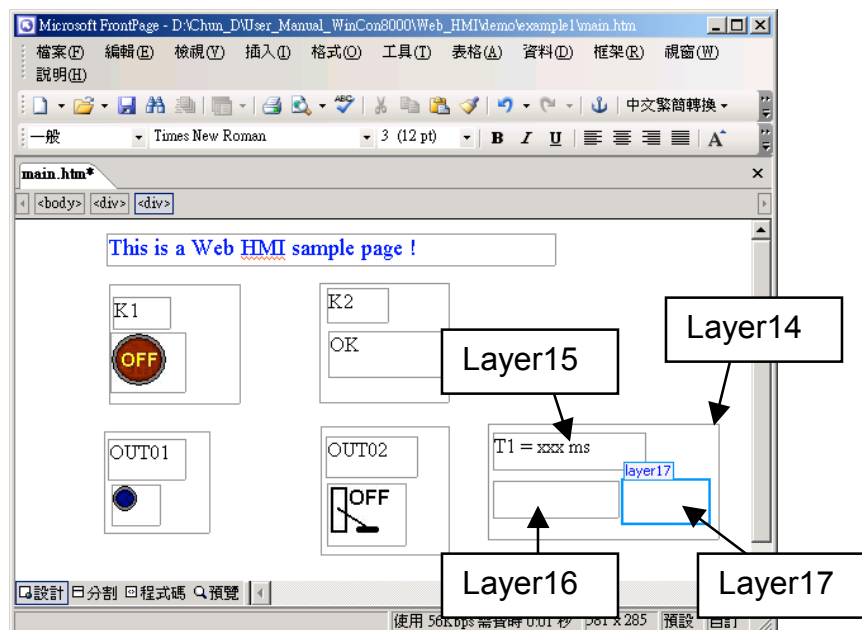
We will use “K1” to display the state of the first input of the I-8077 board, and “K2” for its second input.

Please follow the similar steps to insert “OUT01” & “OUT02” as below. The OUT01 uses “./img/circle_blue0.jpg” as its image source, while OUT02 using “./img/cmd0.jpg”.

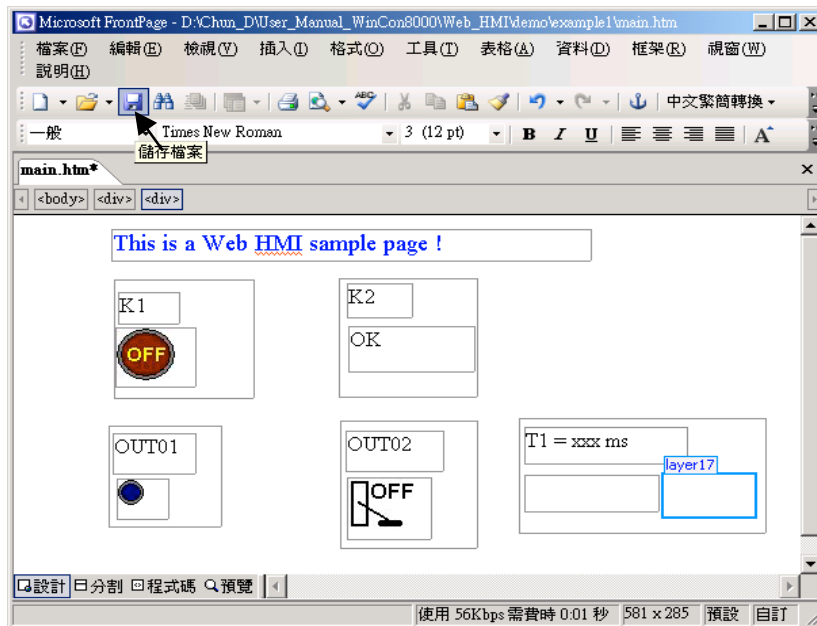


We will use OUT01 to display the state of the first output of the I-8077 board, while “OUT02” is for controlling and displaying the second output of the I-8077.

Now please insert one another “Layer14”. Inside the “Layer14” please insert one “Layer15” with a “T1 = xxx ms” symbol. And two empty Layers – “Layer16” & “Layer17” just below the “Layer15”. We will use T1 to display the Timer value “T1” in the ISaGRAF project.



Click on “Save” to save this page.



4.4.3: Step 3 – Adding Control Code To The Main.htm

Please switch the window to the source code. A valid HTML document will contain the basic objects as below.

If you want to know more about the Web HMI's source code, please refer to Chapter 5.

```
<html>
<title>Your Title here</title>
<head>
<SCRIPT LANGUAGE="JavaScript">
</SCRIPT>
</head>
<body>
</body>
</html>
```

JavaScript code is normally placed inside the “head” area.

The “body” area describes the behavior of this page.

Please go to the body area and then modify the code as below.

Caption Area: Layer1

A Layer is starting with "<div " & ending with "</div>" tag

```
<!-- Caption -->
<font color="blue" size="4">
<div style="position: absolute; width: 353px; height: 24px; z-index: 1; left: 73px; top: 12px" id="layer1">
This is a Web HMI sample page !</div>
</font>
```

K1 Area: Layer2 to Layer4

```
<div style="position: absolute; width: 102px; height: 93px; z-index: 2; left: 75px; top: 52px" id="layer2">
<div style="position: absolute; width: 44px; height: 24px; z-index: 1; left: 3px; top: 10px" id="layer3">
K1</div>
<div style="position: absolute; width: 58px; height: 46px; z-index: 2; left: 1px; top: 38px" id="layer4">
</div>
<p>&nbsp;</p></div>
```

Please insert name="B11" just after the "<img "

K2 Area: Layer5 to Layer7

```
<div style="position: absolute; width: 101px; height: 93px; z-index: 3; left: 241px; top: 51px" id="layer5">
<div style="position: absolute; width: 47px; height: 26px; z-index: 1; left: 6px; top: 4px" id="layer6">
K2</div>
<div style="position: absolute; width: 92px; height: 35px; z-index: 2; left: 7px; top: 38px" id="layer7">
```

```
<font id="font_B12" color="blue" size="3">
```

```
<b id="B12"> OK </b>
```

```
</font> </div>
```

```
<p>&nbsp;</p></div>
```

Please modify "OK <div>" to become

```
<font id="font_B12" color="blue" size="3">
<b id="B12"> OK </b>
</font> </div>
```

OUT01 Area: Layer8 to Layer10

```
<div style="position: absolute; width: 82px; height: 79px; z-index: 4; left: 71px; top: 168px" id="layer8">
<div style="position: absolute; width: 60px; height: 31px; z-index: 1; left: 3px; top: 6px" id="layer9">
OUT01</div>
<div style="position: absolute; width: 37px; height: 31px; z-index: 2; left: 6px; top: 42px" id="layer10">
</div>
<p>&nbsp;</p></div>
```

Please insert name="B1" just after the "<img "

OUT02 Area: Layer11 to Layer13

```
<div style="position: absolute; width:100px; height:100px; z-index: 5; left:242px; top:164px" id="layer11">
<div style="position: absolute; width: 71px; height: 31px; z-index: 1; left: 4px; top: 8px" id="layer12">
OUT02</div>
<div style="position: absolute; width: 61px; height: 48px; z-index: 2; left: 5px; top: 45px" id="layer13">
</div>
```

```
<form name="form_B2" method="post" action="./main.dll">
  <input name="BEGIN" type="hidden">
  <input name="B2" type="hidden" value="0">
  <input name="END" type="hidden">
</form>
```

```
<p>&nbsp;</div>
```

Please insert
Style="cursor:hand" name="B2" onclick="ON_OFF(form_B2, form_B2.B2,
boolean_val[2])"
just after the "<img " tag

Please insert

```
<form name="form_B2" method="post" action="./main.dll">
  <input name="BEGIN" type="hidden">
  <input name="B2" type="hidden" value="0">
  <input name="END" type="hidden">
</form>
```

T1 Area: Layer14 to Layer17

```
<div style="position: absolute; width: 181px; height: 90px; z-index: 6; left: 374px; top: 162px" id="layer14">
<div style="position: absolute; width: 119px; height: 28px; z-index: 1; left: 4px; top: 7px" id="layer15">
```

T1 = <b id="T1">xxx ms</div>

Please modify "T1 = xxx ms </div>" to become
T1 = <b id="T1">xxx ms</div>

```
<div style="position: absolute; width: 98px; height: 28px; z-index: 2; left: 4px; top: 45px" id="layer16">
```

```
<form name="form_L21" method="post" action="/main.dll">
  <input name="BEGIN" type="hidden">
  <input name="L21" type="text" size="8" value="xxx">
  <input name="END" type="hidden">
</form>
```

Please insert below code inside "Layer16"

```
&nbsp;</div>
```

```
<form name="form_L21" method="post" action="/main.dll">
  <input name="BEGIN" type="hidden">
  <input name="L21" type="text" size="8" value="xxx">
  <input name="END" type="hidden">
</form>
```

```
<div style="position: absolute; width: 67px; height: 33px; z-index: 3; left: 106px; top: 44px" id="layer17">
```

<input type="button" value="Enter" onclick="Check_L21()">

```
&nbsp;</div>
<p>&nbsp;</div>
```

Inside the "Layser17", please insert
<input type="button" value="Enter" onclick="Check_L21()">

We have finished the code in the <body> </body> area.

Now please go to the "head" area

In the “head” area, please modify the sample code to be as below.

```
// variable to record object's blink state, 0:not blink, 1: blink, For example:
```

```
// *****
```

```
var B12_blink=0; // init as 0:not blink
```

```
// *****
```

```
// function to blink object
```

```
var blink_step=0;
```

```
function blink_obj()
```

```
{
```

```
  if(blink_step==1)
```

```
  {
```

```
    blink_step=0;
```

```
    // display your object here
```

```
    // blink B12, For example:
```

```
    // *****
```

```
    if(B12_blink==1)
```

```
    {
```

```
      B12.innerText="Error !" ;
```

```
      font_B12.color="red";
```

```
    }
```

```
    // *****
```

```
  }
```

```
  else
```

```
  {
```

```
    blink_step=1;
```

```
    // un-display your object here
```

```
    // blink B12, For example:
```

```
    // *****
```

```
    if(B12_blink==1)
```

```
    {
```

```
      B12.innerText="" ;
```

```
      font_B12.color="red";
```

```
    }
```

```
    // *****
```

```
  }
```

```
  setTimeout("blink_obj()", blink_period);
```

```
}
```

The “Error !” symbol will blink when the K2 = True in this example. Please un-mask the code inside these 3 areas.

We need a function "Check_L21 to check the entered T1 value and post it to the Wincon.
Please un-mask the sample code to be as below.

```
// form sample, to check value of L21 & then post val to controller
// For example:
// *****
```

```
function Check_L21()
{
  var val=form_L21.L21.value;
  if(val>12000 || val<4000)
  {
    alert("T1's value should be in the range of 4000 to 12000");
    return;
  }
  Check(form_L21); // post value to the controller
}
```

```
// *****
```

And also inside the "refresh_data()" " function, please insert below code.

```
// To refresh displayed data, this function is called by IE about every 1.5 sec later
```

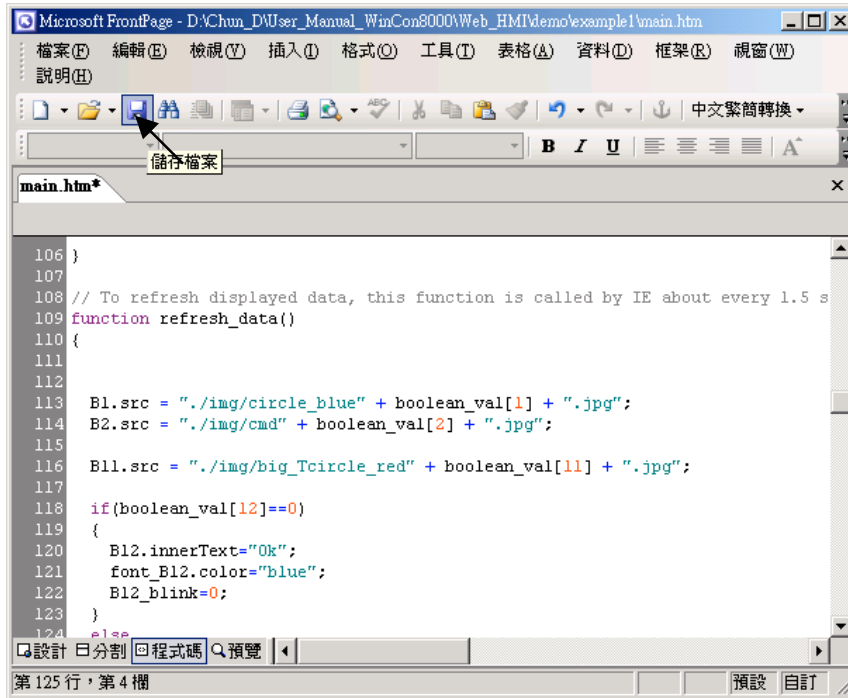
```
function refresh_data()
{
  B1.src = "./img/circle_blue" + boolean_val[1] + ".jpg";
  B2.src = "./img/cmd" + boolean_val[2] + ".jpg";

  B11.src = "./img/big_Tcircle_red" + boolean_val[11] + ".jpg";

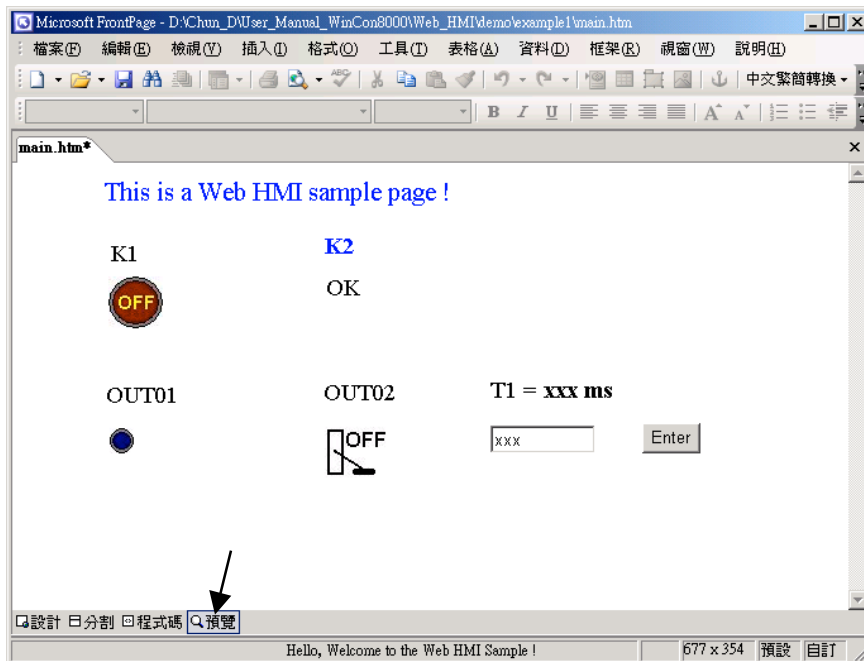
  if(boolean_val[12]==0)
  {
    B12.innerText="Ok";
    font_B12.color="blue";
    B12_blink=0;
  }
  else
  {
    B12_blink=1;
  }

  T1.innerText=timer_val[21] + " ms";
}
```


Now we have finished all the code. Please save it.



You may click on "Preview" to simulate its run time behavior.



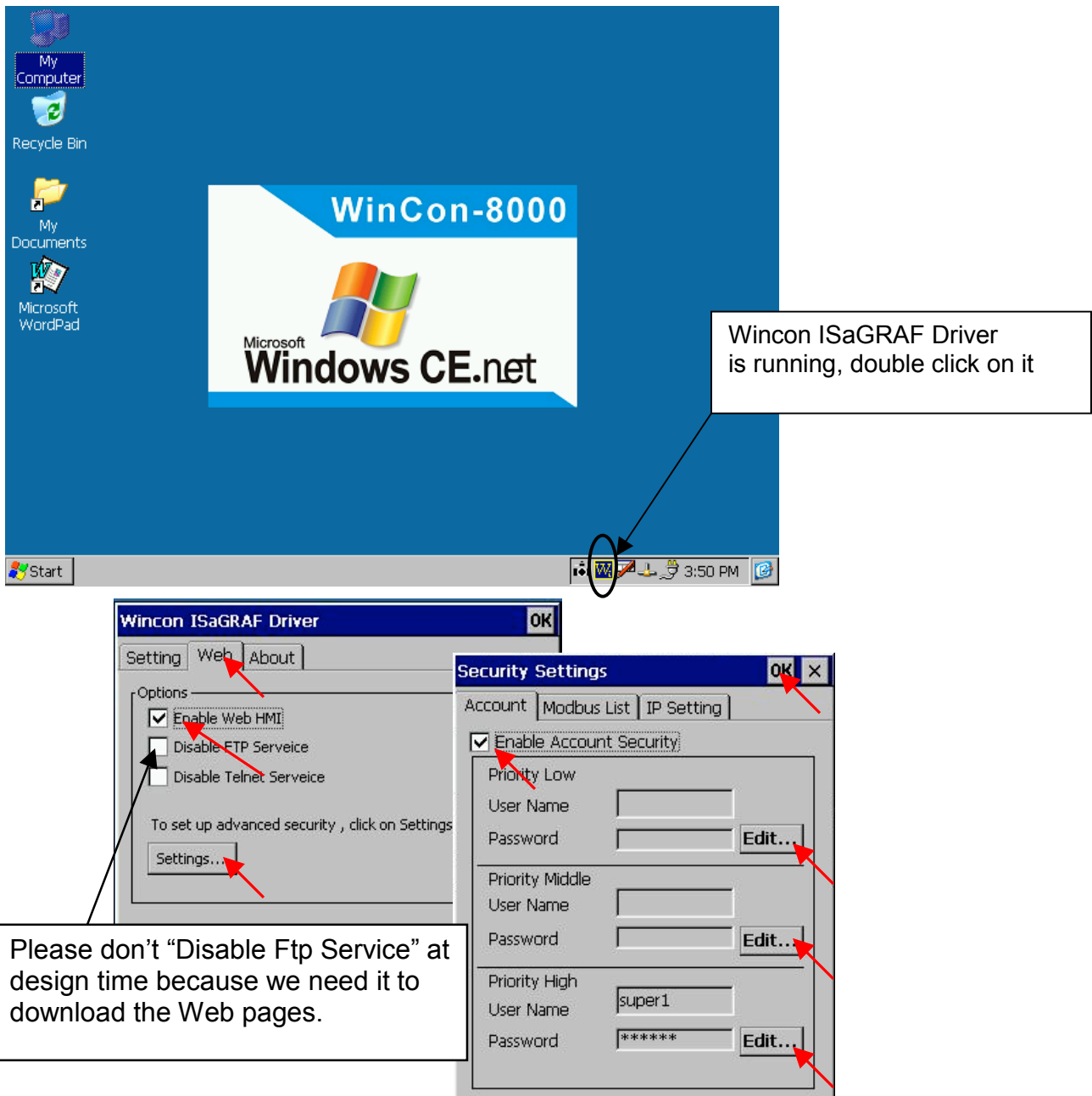
4.4.4: Step 4 – Download Web HMI Pages To The Controller

The steps are similar as listed in Section 3.2. If you haven't practiced "Setting Up A Web HMI Demo" listed in the Chapter 3, it's better to do it once to get familiar with it.

First set the web options.

Check on "Enable Web HMI" and then click on "Setting", Please check on "Enable Account Security" and then click on "Edit" to set (username , password). **Then remember to click on "OK"**

Note: If "Enable Account Security" is not check, any user can easily get access to your Wincon through the Internet Explorer.



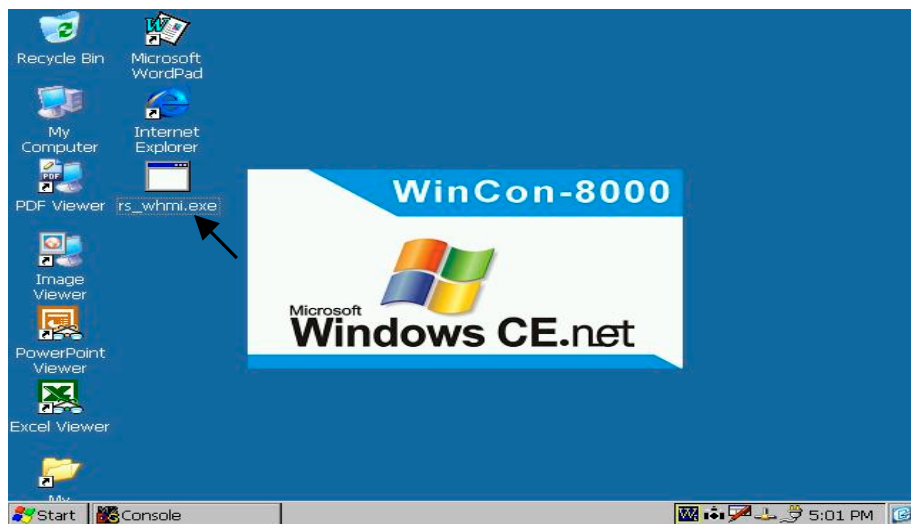
And then, please copy all files in this example1 to the controller

<your hard drive>:\example1\ *.*

to the Wincon-8037/8337/8737's

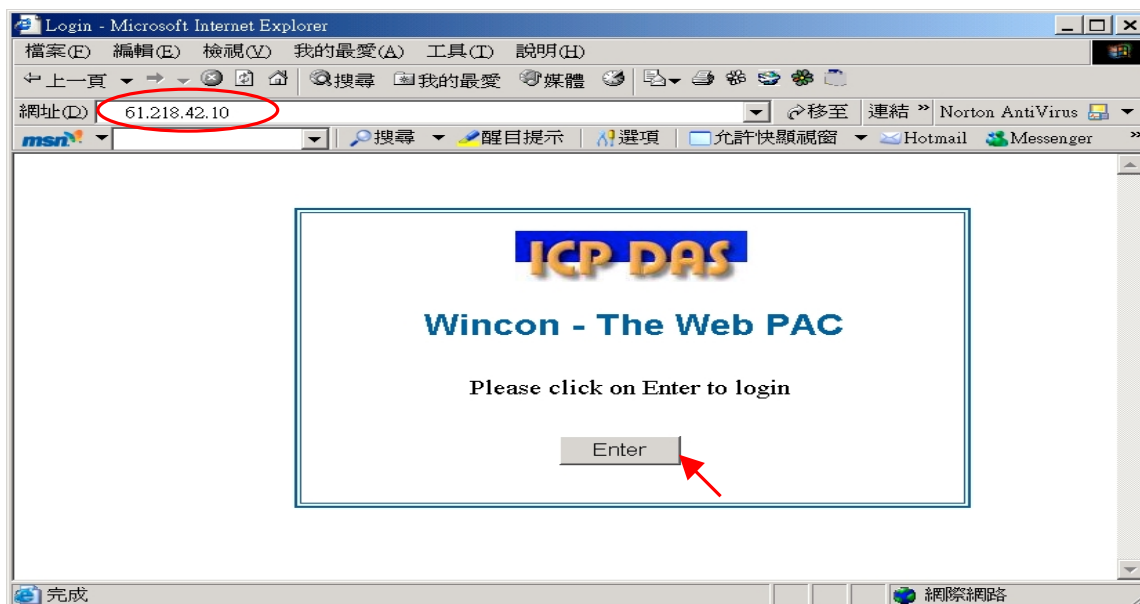
Compact Flash\Temp\HTTP\WebHMI\

Since the Web Pages are modified or new copied, please run "rs_whmi.exe" to reset the Web server. **The "rs_whmi.exe" must be run every time when user has modified any file in the W-8x37's Compact Flash\Temp\HTTP\WebHMI**

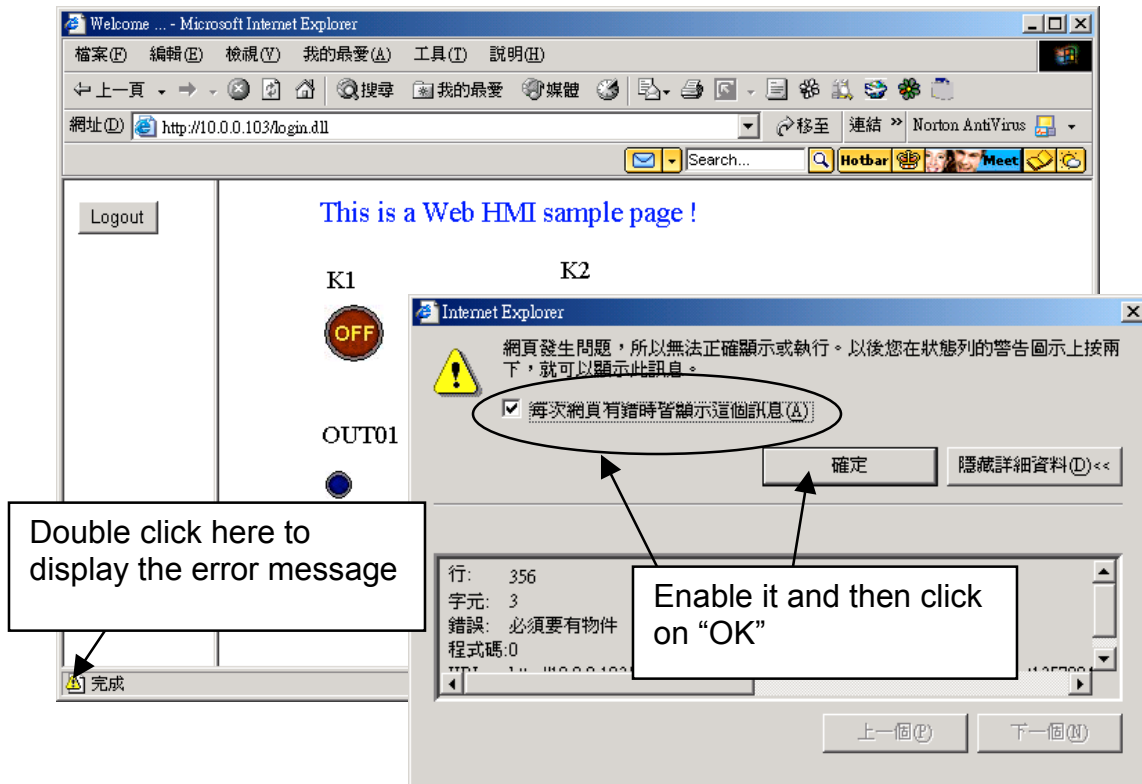


Show Time:

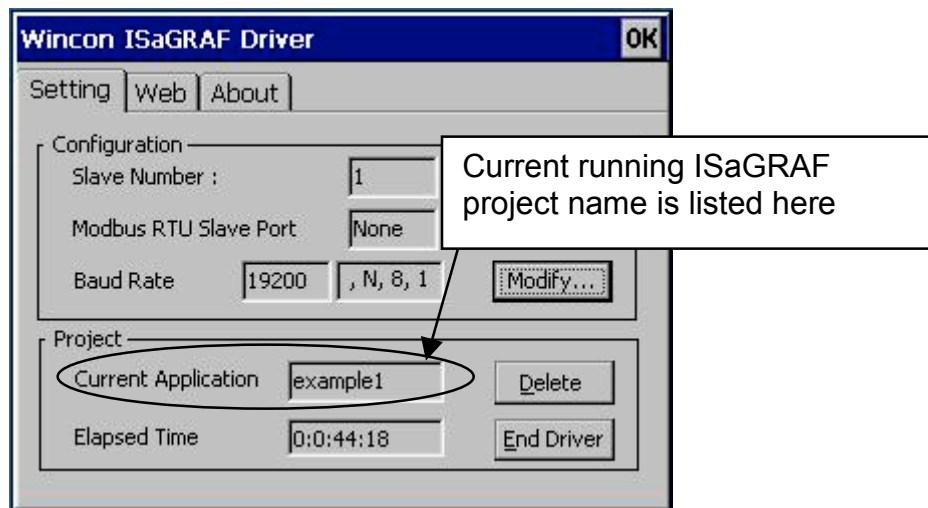
Please run Internet Explorer (Rev. 6.0 or higher), key in the IP address of your W-8037/8337/8737. For example: 61.218.42.10 or http://61.218.42.10



If there is something wrong with the web page. You may enable the below item to display the debug message every time it has error.



And also check if your ISaGRAF project already download to the controller (Section 4.3 or or section 3.2.4).



And do you assign the correct Modbus Network address to the respective ISaGRAF variables ? (Section 4.1.5).

Chapter 5: Web HMI Basics

Note:

1. This chapter describes the programming basics for the Web HMI. We will not focus on the HTML programming basics. If you want to know more about the HTML programming, the best way is to “buy a HTML related book” from the bookstore. There are a lot of books doing this job.
2. The Web HMI only supports the basic HTML tags. It doesn't support ASP, PHP or JSP or other Page Server language.
3. Please do not use `<frameset>` `</frameset>` , `<frame>` `</frame>` in the Web HMI.
4. Note: The object name & object ID & code & variable name & function name is case sensitive. For example, `refresh_data()` and `Refresh_data()` is different.
5. There are more than 10 Web HMI examples in the Wincon-8x37's CD-ROM. Please refer to section 3.1.

5.1: Basic Files For The Web HMI

The basic Web HMI files includes 2 folders and 3 DLL files and 4 htm files as below.

<code>./img/</code>	(default image files - *.jpg , *.bmp , *.gif)
<code>./msg/</code>	(default message files – wincon.js & xxerror.htm)
<code>whmi_filter.dll</code> <code>login.dll</code> <code>main.dll</code>	(three DLL files)
<code>index.htm</code>	(first default page)
<code>login.htm</code>	(the Web HMI welcome page)
<code>menu.htm</code>	(the page-menu page, normally on the left on the Internet Explorer)
<code>main.htm</code>	(first page when successfully login)

User may put his own image files into the folder named as “user_img”. And put user-defined javascript file or css file into the folder named as “user_msg”. Other folder name is not acceptable by the Wincon Web HMI.

The “index.htm” file is the default entry page of the web server. User should not modify it. The “index.htm” re-directs to the “login.htm” file in 1 to 2 second when someone visits the Wincon-8x37 via the Internet Explorer.

User may modify the “login.htm” , “menu.htm” & “main.htm” to fit his own need.

5.2: Login.htm

Login.htm is the first welcome page when a user visiting. It can be modified. Below is the basic code for the login.htm

```
<html>
<head>

<title>Login</title>

<meta http-equiv=pragma content=no-cache>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" >

<script language="JavaScript">
var random_val=123;
function get_random_val()
{
    var rightNow = new Date();
    random_val += 323456789*rightNow.getMinutes() +
    107654321*(rightNow.getTime()%1000);
    setTimeout("get_random_val()", 197); // repeat call
}

//check if username and password are empty
function validate(fm)
{
    setKey(fm);
    return true;
}

//Embed key while submitting
function setKey(fm)
{
    var rightNow = new Date();
    cookieVal = random_val+rightNow.getTime();
    fm.key_.value = cookieVal;
}
</script>

</head>

<body onload="get_random_val()">

<div style="position: absolute; width: 332px; height: 34px; z-index: 5; left: 147px; top: 27px"
id="layer1">
Welcome !</div>
```

This line is only for the "Login.htm" , please do not apply to other pages. For example, the "menu.htm" & "main.htm" & other .htm pages.

Please apply your charset here.
For example,
English: UTF-8
Chinese: gb2312
Traditional Chinese: big5
or other language

get_random_val() should be always called at the beginning of the Login.htm . It is the entry point of the Login.htm

Your caption here.

```
<div style="position:absolute; width:122px; height:38px; z-index:4; left: 171px; top: 95px;" id="layer2">
```

```

    <form name="form1" action="./login.dll" method="post">
        <input type="hidden" name="key_">
        <input type="submit" name="Submit" value=" Enter " style="cursor:hand" onClick="return
validate(this.form)">
    </form>

```

"form1" is necessary

You may modify " Enter " to your own word. For example "請進". This may require to modify the related charset at the beginning of this page.

```
</div>
```

```
</body>
```

```
<!-- To ensure no-cache work -->
```

```
<head>
```

```
<meta http-equiv=pragma content=no-cache>
```

```
</head>
```

```
</html>
```

This code is only for the "Login.htm", please do not apply to other pages. For example, the "menu.htm" & "main.htm" & other .htm pages.

That's all the login.htm need. You can insert more images or text to it. Only remember to keep its basic code.

5.3: menu.htm

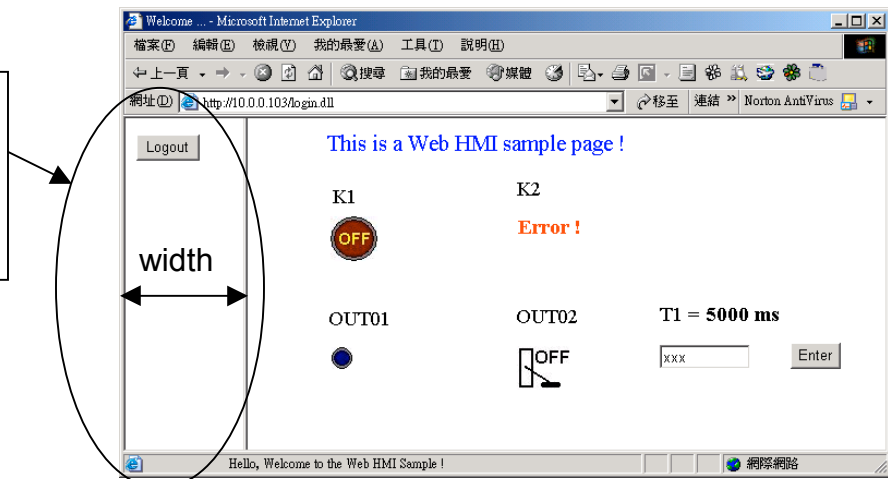
Note:

If you want to know more about the multi-page application, there are two demos in the Wincon-8x37 CD-ROM: \napdos\isagraf\wincon\WebHMI_Demo\demo_05 & demo_05a . Demo_05 place its page-menu on the left, while Demo_05a on the top.

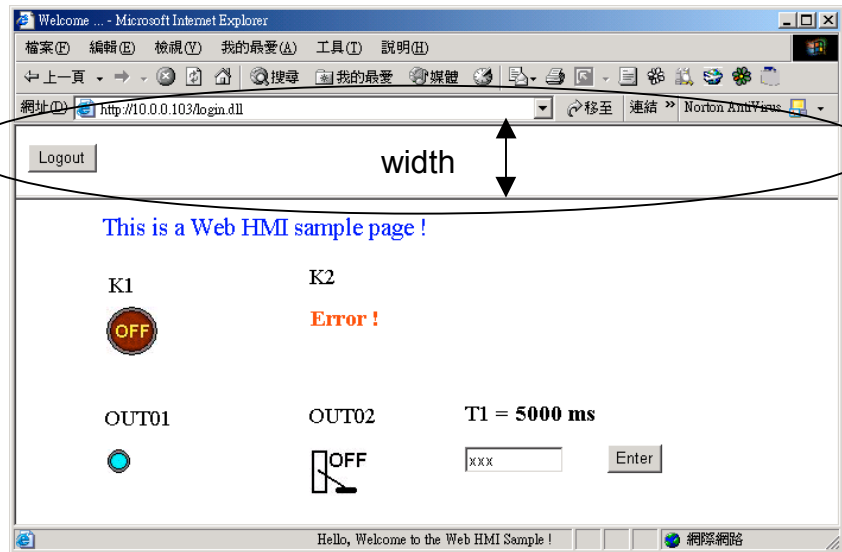
Menu.htm defines the Page-menu of the Web HMI especially for the multi-page application. The page-menu can place only on the left or on the top.

On the left.

The width & scrolling can be modified.



On the top.
The width & scrolling can be modified.



Below is the basic code for the menu.htm

```
<!-- top_or_left=1 , scrolling=0 , width=60 , resize=1 -->
```

The first line is not a comment, it defines the Page-Menu behavior
top_or_left: 1:Top , 0:Left
scrolling: 1:Yes , 0:No
width: width of the Menu Frame, 0 – 999 (unit is pixel)
resize 1:Yes , 0:No

```
<html>
<head>
<title>Title1</title>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" >
```

```
<SCRIPT LANGUAGE="JavaScript" src="./msg/wincon.js"></SCRIPT>
```

```
<SCRIPT LANGUAGE="JavaScript">
```

```
function start1()
{
  A_11();
}
function refresh_data()
{
  if(run_at_pc==1) return;
}
</SCRIPT>
```

This line is necessary
for menu.htm ,
main.htm & other
multi-pages

Please apply your charset here.
For example,
English: UTF-8
Chinese: gb2312
Traditional Chinese: big5
or other language

```
</head>
```

```
<body onload="start1()">
```

start1() is the entry point of the menu.htm


```

<!-- Logout button -->
<form name="form_logout" method="post" action="./login.dll">
  <input style="cursor:hand" name="CMD" type="submit" value="Logout" onClick="return
logout(this.form)">
</form>

</body>
</html>

```

form_logout is for the logout button.

Note:

If you want to know more about the multi-page application, there are two demos in the Wincon-8x37 CD-ROM: \napdos\isagraf\wincon\WebHMI_Demo\demo_05 & demo_05a . Demo_05 place its page-menu on the left, while Demo_05a on the top

5.4: main.htm

5.4.1: A Simple Main.htm Example

Before going further in the main.htm, first take a look at a simple main.htm example. This example only display a "Hello !" message when successfully login, nothing else.

```

<html>
<head>
<title>Title1</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" >

<SCRIPT LANGUAGE="JavaScript" src="./msg/wincon.js"></SCRIPT>

<SCRIPT LANGUAGE="JavaScript">
show_scroll_word(200,"Hello, Welcome to the Web HMI Sample !");

function refresh_data()
{
}

</SCRIPT>
</head>

<body onLoad="init()">

<font color="blue" size="4">
<div style="position: absolute; width: 353px; height: 24px; z-index: 1; left: 73px; top: 12px"
id="layer1"> Hello !</div>
</font>

</body>
</html>

```

Please apply your charset here. For example, English: UTF-8
Chinese: gb2312 , Traditional Chinese: big5 , or other language

This line is necessary for menu.htm , main.htm & other multi-pages

Calling show_scroll_world() will display a moving word at the bottom of the Internet Explorer. Here 200 means 200 ms. You may make it slower, for example, using 500.

refresh_data() is called when the Internet Explorer has received the requested data from the controller. It is called in the period about 1.25 to 5 seconds depends on the communication quality.

init() is the entry pint of the main.htm & other multi-pages.

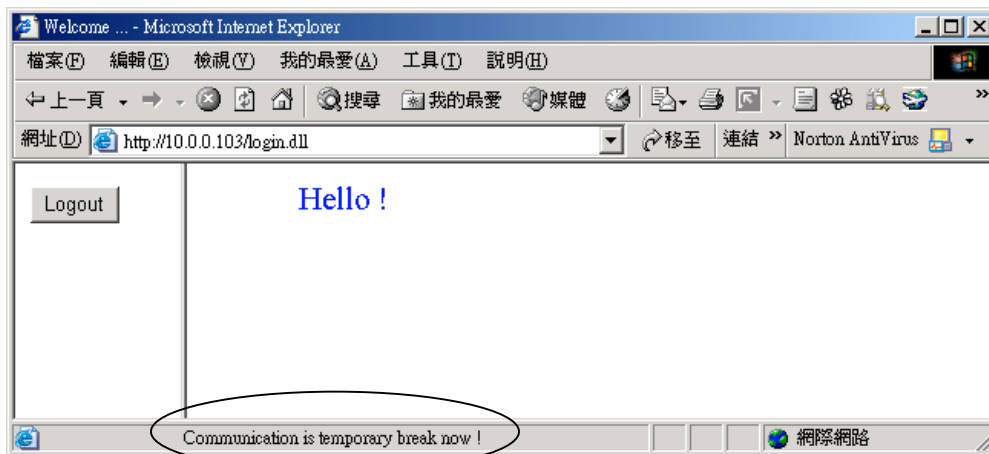
A layout object is starting with "<div" & ending at "</div>" tags. Here only show a message "Hello !"

You may replace the main.htm in the Wincon-8x37 CD-ROM:

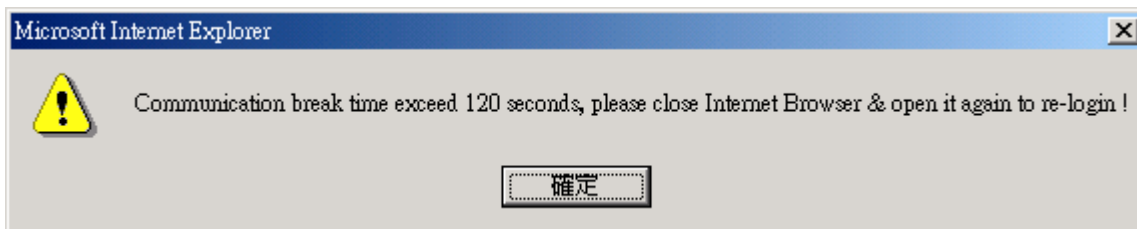
\\napdos\isagraf\wincon\WebHMI_Demo\sample to the above main.htm & download it to the controller (refer to section 4.4.4). You will see the below window when you login successfully.



User may try to plug out the Ethernet cable of the Wincon or of your PC. You will see it show "Communication is temporary break now !" in about 10 seconds. When you plug the cable back, the communication will be recovered in about 10 to 45 seconds.



If the communication break time exceeds 120 seconds, it will show the below message. You have to close the Internet Explorer & open it again to re-login.



5.4.2: More About The refresh_data() Function And Dynamic Data

Note: The code & variable name & function name is case sensitive. For example, refresh_data() is correct, however Refresh_data() is not correct.

The refresh_data() function must always apply in the main.htm and other multi-pages. It is called when the Internet Explorer has received the requested data from the controller. The calling period is about 1.25 to 5 seconds depends on the communication quality

The refresh_data() is often used for refreshing the dynamic data. For example, the boolean value , integer value, timer value or float value of the variables in the ISaGRAF project.

The Internet Explorer can access to the data in the ISaGRAF project only when they are assigned a unique Modbus Network Address No (refer to section 4.1.5). The Web HMI only accepts Network Address No in the range of 1 to 1024. The data without a Network Address No (No. = 0) or not in the range of (1 to 1024) is not accessible by the Internet Explorer.

The main.htm and other multi-pages can use the below variable array to access to the ISaGRAF's data (case sensitive). The identifier appeared in the [] is the related Network Address No. For example boolean_val[2] means the boolean value of the ISaGRAF boolean data which is assigned with the Network Address No. = 2.

boolean_val	boolean value in the ISaGRAF
word_val	word value in the ISaGRAF, -32768 to +32767
float_val	real value in the ISaGRAF, for ex, 1.234 , -0.456E-02
timer_val	timer value in the ISaGRAF, unit is ms, max = 86399999 (< 1 day)
string_val	message value in the ISaGRAF, max string length is 255

To access to long integer value (32-bit integer) please use get_long_val() function. For example, get_long_val(11) , get_long_val(13) , get_long_val(15).

get_long_val()	long integer value in the ISaGRAF, -2147483648 to +2147483647
-----------------	---

Note:

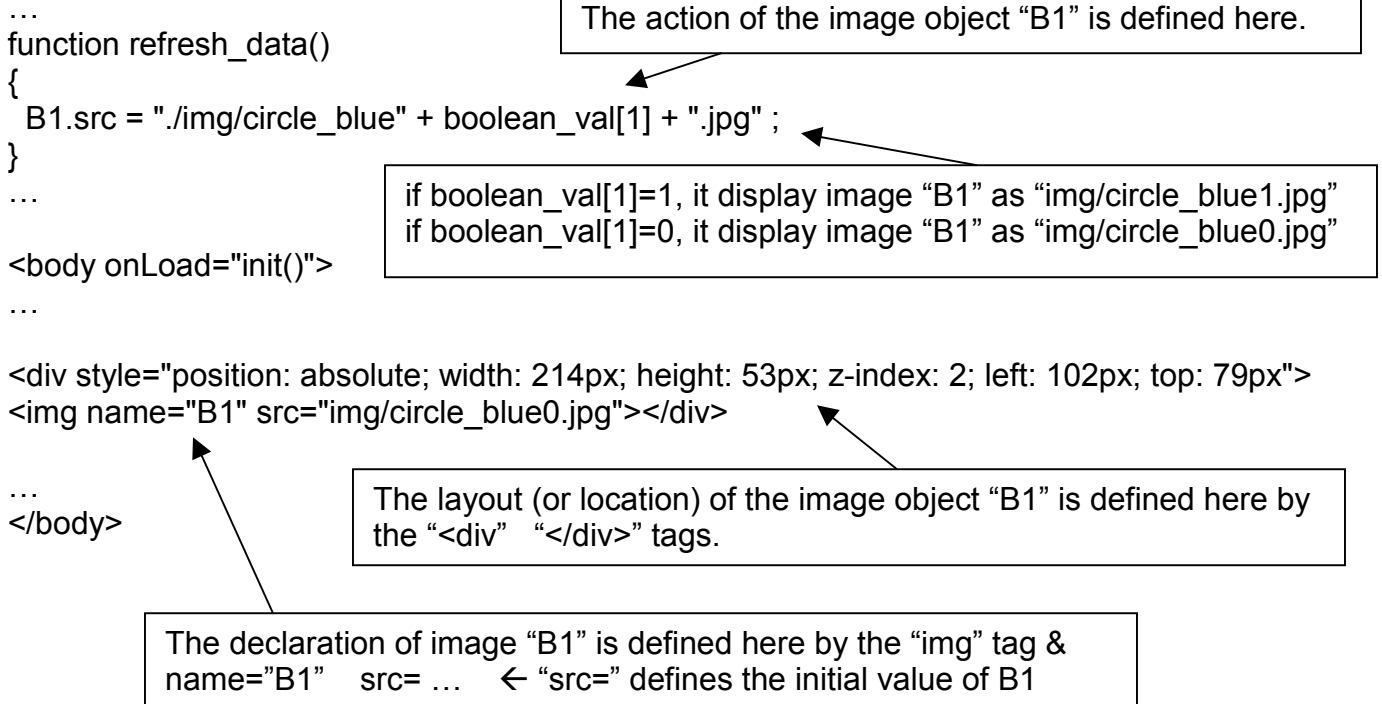
The long integer & timer & float variable's Network Address No. must occupy 2 No. in the ISaGRAF project (refer to section 4.2 of "User's Manual of ISaGRAF Embedded Controllers" or in the CD-ROM:\napdos\isagraf\wincon\english_manu\ "User_Manual_I_8xx7.pdf").

That means if you assign a Network Address No.= 11 to a Real type variable(or Timer or integer will have 32-bit value – larger than 32767 or smaller than -32768), the next No. 12 should not assigned to any other variable in the ISaGRAF project. However you may assign No.=13 to one another variable.

5.4.2.1: Displaying Dynamic Boolean Data

Demo example: whmi_02 & whmi_05 (section 3.1)

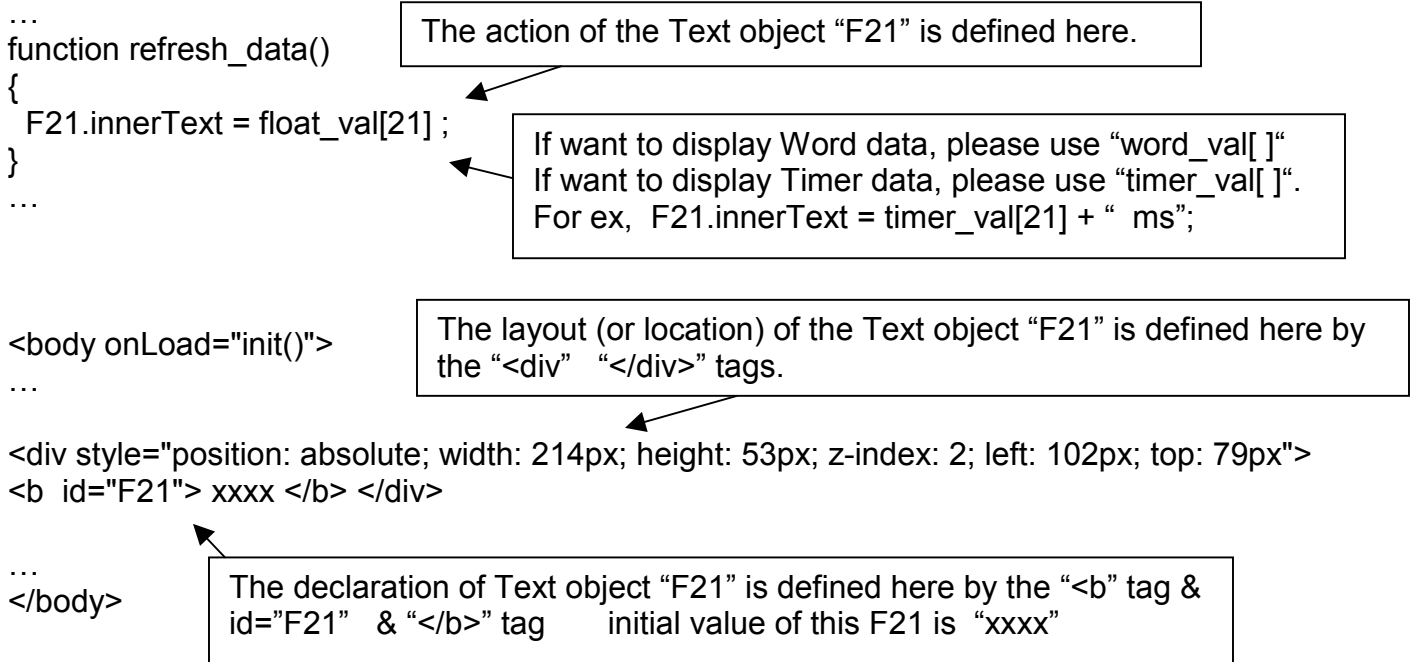
Let's look back to the refresh_data function. If user want to display the dynamic boolean value, the below code can be used.



5.4.2.2: Displaying Dynamic Float & Word & Timer Data

Demo example: whmi_01 & whmi_03 & whmi_05 (section 3.1)

If user want to display the dynamic float value, the below code can be used.



5.4.2.3: Displaying Dynamic Long Integer Data

Demo example: whmi_03 & whmi_05 (section 3.1)

If user want to display the dynamic long integer value (32-bit format), the below code can be used.

```
...
function refresh_data()
{
  L11.innerText = get_long_val(11) ;
}
...
```

The action of the Text object "L11" is defined here.

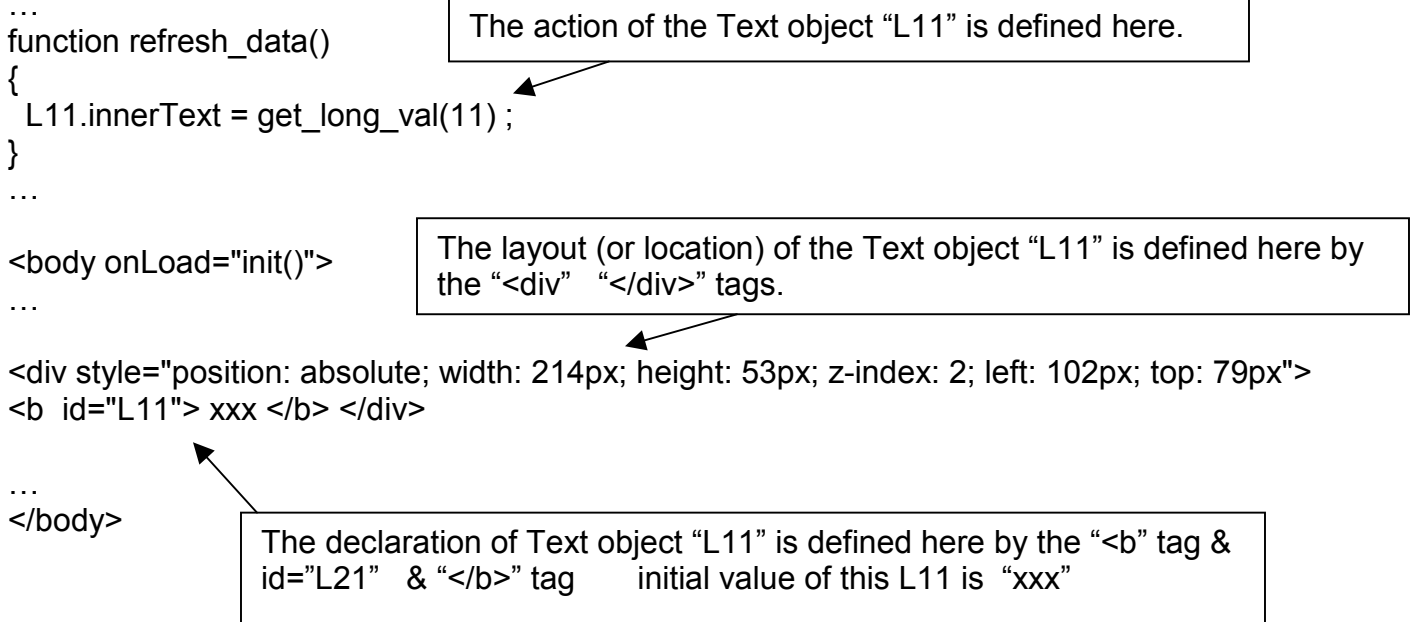
```
<body onLoad="init()">
...
```

The layout (or location) of the Text object "L11" is defined here by the "<div" "</div>" tags.

```
<div style="position: absolute; width: 214px; height: 53px; z-index: 2; left: 102px; top: 79px">
<b id="L11"> xxx </b> </div>

...
</body>
```

The declaration of Text object "L11" is defined here by the "<b" tag & id="L21" & "" tag initial value of this L11 is "xxx"

The diagram illustrates the code for displaying dynamic long integer data. It shows three code snippets: a JavaScript function, an HTML body tag, and a div/b tag. Callout boxes with arrows point to specific parts of the code: the first box points to the L11.innerText assignment in the JavaScript function; the second box points to the div style and the b id="L11" tag in the HTML; the third box points to the b tag and its content in the HTML.

5.4.2.4: Displaying Dynamic String Data

If user want to display the dynamic string value (max length is 255), the below code can be used.

```
...
function refresh_data()
{
  S31.innerText = string_val[31] ;
}
...
```

The action of the Text object "S31" is defined here.

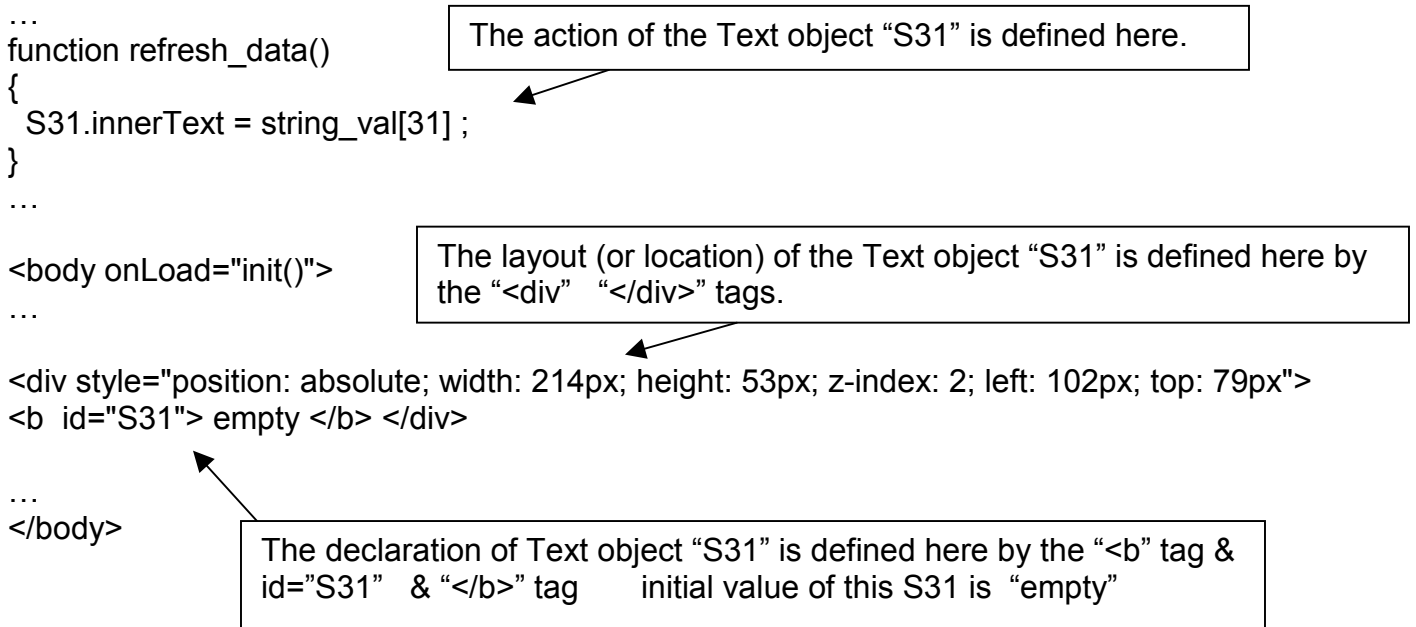
```
<body onLoad="init()">
...
```

The layout (or location) of the Text object "S31" is defined here by the "<div" "</div>" tags.

```
<div style="position: absolute; width: 214px; height: 53px; z-index: 2; left: 102px; top: 79px">
<b id="S31"> empty </b> </div>

...
</body>
```

The declaration of Text object "S31" is defined here by the "<b" tag & id="S31" & "" tag initial value of this S31 is "empty"

The diagram illustrates the code for displaying dynamic string data. It shows three code snippets: a JavaScript function, an HTML body tag, and a div/b tag. Callout boxes with arrows point to specific parts of the code: the first box points to the S31.innerText assignment in the JavaScript function; the second box points to the div style and the b id="S31" tag in the HTML; the third box points to the b tag and its content in the HTML.

5.4.2.5: Trigger A Boolean Object To Blink

Demo example: whmi_02 & whmi_05 (section 3.1)

Some application may need a message to blink when the boolean value changes. For example, If `boolean_val[12]` is False, it means "OK". However if `boolean_val[12]` is True, it means "Error !". User may want to make this "Error !" blink to attract viewer's attention.

The below code can do this job.

```
...
var blink_period=500;
setTimeout("blink_obj()", blink_period);
var B12_blink=0; // init as 0:not blink
var blink_step=0;

function blink_obj()
{
  if(blink_step==1)
  {
    blink_step=0;
    if(B12_blink==1)
    {
      B12.innerText="Error !" ;
      font_B12.color="red";
    }
  }
  else
  {
    blink_step=1;
    if(B12_blink==1)
    {
      B12.innerText="" ;
      font_B12.color="red";
    }
  }
  setTimeout("blink_obj()", blink_period);
}
...
```

The blinking period, unit is ms

Setup a timer to handle the blinking action

1: to blink , 0: not blink

Blink step 1:
To display "Error !" in red color.

Blink step 2:
To display "" (nothing) in red color.

```
function refresh_data()
{
```

```
  if(boolean_val[12]==0)
  {
    B12.innerText="Ok";
    font_B12.color="blue";
    B12_blink=0;
  }
  else
  {
    B12_blink=1;
  }
}
```

The action of the Text object "B12" is defined here. If boolean_val[12]=0, not blink. However If boolean_val[12]=1, blink.

```
...
<body onLoad="init()">
```

The layout (or location) of the Text object "B12" is defined here by the "<div" "</div>" tags.

```
<div style="position: absolute; width: 214px; height: 53px; z-index: 2; left: 102px; top: 79px">
```

```
<font id="font_B12" color="blue" size="3">
```

```
<b id="B12">OK</b>
```

The "" & "" tags can be used for controlling the font's color and font's size.

```
</font>
```

```
</div>
```

The declaration of Text object "B12" is defined here by the "<b" tag & id="B12" & "" tag initial value of this B2 is "OK"

```
...
</body>
```

5.4.2.6: Displaying Float Value With Fixed Digit Number Behind The "." Symbol

Demo example: whmi_06 & whmi_07 (section 3.1)

The float_str1(para1 , para2) function can convert float value to a string with fixed digit number behind the dot "." symbol

para1 is the float value to be converted, for ex, 1.234567

para2 is the digit number behind the "." dot symbol, 0 to 6

for ex, float_str1(1.234567, 3) return "1.234", while float_str1(1.234567, 2) return "1.23"

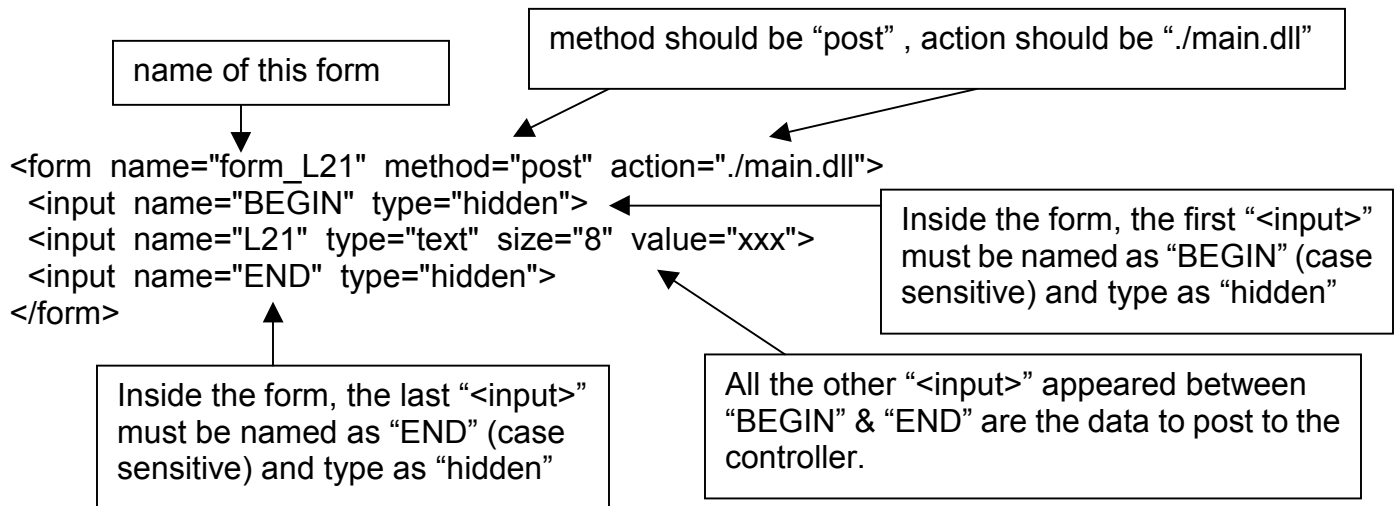
```
...
function refresh_data()
{
  F21.innerText = float_str1( float_val[21] , 3) ;
}
```

Convert float val at Network Address 21 to a string with digit number = 3 behind the "." dot symbol

5.4.3: Post Data To The Controller

The former section 5.4.2 listing how to get and display data from the controller. This section focuses on posting data to the controller, in other word to control the Wincon via the Internet Explorer.

To set a new value to the boolean, word, long integer, float , timer and string variables in the ISaGRAF project, we need “form” object appeared in the main.htm or other multi-pages. A “form” object looks like as below.



The “<input>” name to control the Wincon’s data must follow below format. The number follows behind the first letter should be in the range of 1 to 1024. This number point to the variable name in the ISaGRAF project with the same Modbus Network Address No.

B	point to the ISaGRAF boolean data , for ex, B5 , B109
W	point to the ISaGRAF word data (-32768 to +32767), for ex, W9 , W1001
L	point to the ISaGRAF long integer data (-2147483648 to +2147483647), for ex, L21 This “L” Also point to the ISaGRAF timer data
F	point to the ISaGRAF real data, for ex, F13 , F235
S	point to the ISaGRAF message data , for ex, S18

Note:

The long integer & timer & float variable’s Network Address No. must occupy 2 No. in the ISaGRAF project (refer to section 4.2 of “User’s Manual of ISaGRAF Embedded Controllers” or in the CD-ROM:\napdos\isagraf\wincon\english_manu\“User_Manual_I_8xx7.pdf”).

That means if you assign a Network Address No.= 11 to a Real type variable(or Timer or integer will have 32-bit value – larger than 32767 or smaller than -32768), the next No. 12 should not assigned to any other variable in the ISaGRAF project. However you may assign No.=13 to one another variable.

5.4.3.1: Post Boolean Value to The Controller

A. To post by the image

```
...
function ON_OFF(form_obj, obj, current_boo_value)
{
  if(current_boo_value==0)
  {
    flag = confirm("turn ON ?");
    if(flag) obj.value=1;
  }
  else
  {
    flag = confirm("turn OFF ?");
    if(flag) obj.value=0;
  }
  if(flag)
  {
    if(GetUserID(form_obj)==true) form_obj.submit();
  }
}
function refresh_data()
{
  B2.src = "img/cmd" + boolean_val[2] + ".jpg" ;
}
...
```

ON_OFF function is used for posting boolean value to the controller by reference to the current boolean value.

The first parameter is the name of the "form".
The second parameter is the "<input>" name inside the form.
The last parameter is the current boolean value.

Demo example: whmi_02 & whmi_05 (section 3.1)

Display the current boolean image. In this example,
0: display "img/cmd0.jpg" , 1: "img/cmd1.jpg"

```
<body onLoad="init()">
...
```

The layout (or location) of the image object "B2" is defined here by the "<div>" "</div>" tags.

```
<div style="position: absolute; width: 100px; height:100px; z-index: 5; left: 242px; top: 164px" >
```

"cursor:hand" will display the mouse arrow as a hand when entering the image area

```

```

Name of the form

Name of the image object

The onclick will call ON_OFF() when the mouse click on it.

The first parameter is the name of the "form". Here is "form_B2"

The second parameter is the "<input>" name inside the form. Here is "form_B2.B2"

The last is the current boolean value. Here is boolean_val[2]

```
<form name="form_B2" method="post" action="./main.dll">
  <input name="BEGIN" type="hidden">
  <input name="B2" type="hidden" value="0">
  <input name="END" type="hidden">
</form>

</div>
...
</body>
```

Name of "<input>" inside the form. Here is "B2". Because it is inside "form_B2", then we must use the name of "form_B2.B2" to identify it.

B.To post by buttons

Demo example: whmi_02 & whmi_05 (section 3.1)

```
...
function ON_(form_obj, obj)
{
  flag = confirm("turn ON ?");
  if(flag)
  {
    obj.value=1;
    if(GetUserID(form_obj)==true) form_obj.submit();
  }
}
```

ON_ function is used for posting boolean value as "True" to the controller .

```
function OFF_(form_obj, obj)
{
  flag = confirm("turn OFF ?");
  if(flag)
  {
    obj.value=0;
    if(GetUserID(form_obj)==true) form_obj.submit();
  }
}
```

OFF_ function is used for posting boolean value as "False" to the controller .

```
function refresh_data()
{
  B2.src = "img/big_Tcircle_red" + boolean_val[2] + ".jpg" ;
}
```

Display the current boolean image. In this example, 0: "img/big_Tcircle_red0.jpg" , 1: "img/ big_Tcircle_red1.jpg"

```
...
<body onLoad="init()">
```

The layout (or location) of the image object "B2" is defined here by the "<div>" "</div>" tags.

```
...
<div style="position: absolute; width: 56px; height:40px; z-index: 5; left: 82px; top: 69px" >

</div>
```

```
<div style="position:absolute; left:85px; top:124px; width:42px; height:27px;">
<input type="button" value="ON" style="cursor:hand" onClick="ON_(form_B2, form_B2.B2)">
<form name="form_B2" method="post" action="/main.dll">
  <input name="BEGIN" type="hidden" value="">
  <input name="B2" type="hidden" value="1">
  <input name="END" type="hidden" value="">
</form>
</div>
```

Name of "<input>" inside the form. Here is "B2". Because it is inside "form_B2", then we must use the name of "form_B2.B2" to identify it.

A button to call ON_()
First parameter is the name of the form. Here is "form_B2"
The second is the name of the "<input>" inside the form. Here is "form_B2.B2"

```
<div style="position:absolute; left:85px; top:166px; width:47px; height:31px">
<input type="button" value="OFF" style="cursor:hand" onClick="OFF_(form_B2, form_B2.B2)">
</div>
...
</body>
```

A button to call OFF_()
First parameter is the name of the form. Here is "form_B2"
The second is the name of the "<input>" inside the form. Here is "form_B2.B2"

5.4.3.2: Post Word & Long & Float & Timer & String Value to The Controller

...

```
function Check(form_obj)
{
  flag = confirm("Are you sure?");
  if(flag)
  {
    if(GetUserID(form_obj)==false) { return false; }
    form_obj.submit();
    return true;
  }
  else
  {
    return false;
  }
}
```

Check() is used for posting any "form".

Demo example:
whmi_03 & whmi_04 & whmi_05 &
whmi_06 & Whmi_07 (section 3.1)

```
function refresh_data()
{
  L15.innerText=get_long_val(15);
  F17.innerText=float_val[17];
}
```

Display dynamic value here.
If data is word , please use word_val[]
If data is timer, please use timer_val[]
If data is string, please use string_val[]

```
<body onLoad="init()">
```

...

The layout (or location) of the text object "L15" & "F17" are defined here by the "<div>" "</div>" tags.

```
<div style="position: absolute; width: 195px; height: 25px; z-index: 2; left: 45px; top: 52px" >
L15 = <b id="L15">xxxx</b></div>
<div style="position: absolute; width: 196px; height: 29px; z-index: 3; left: 45px; top: 82px" >
F17 = <b id="F17">xxxx</b></div>
```

```
<div style="position:absolute; left:47px; top:131px; width:204px; height:60px">
```

```
<form name="form1" method="post" action="/main.dll">
```

```
<input name="BEGIN" type="hidden" value="">
```

```
<input name="L15" type="text" value="Enter long val (L15)">
```

```
<input name="F17" type="text" value="Enter float val (F17)">
```

```
<input name="END" type="hidden" value="">
```

```
</form>
```

```
</div>
```

```
<div style="position:absolute; width:74px; height:31px; left: 234px; top: 150px;">
```

```
<input type="button" style="cursor:hand" onClick="return Check(form1)" value="Enter">
```

```
</div>
```

...

```
</body>
```

"cursor:hand" will display the mouse arrow as a hand when entering the button area

When mouse click on this button, it calls Check() to post to the controller

text input L15 & F17 inside the "form1" if data is timer, please use "L". And "W" for word. "S" for string

5.5: Multi-Pages

The Web HMI in the Wincon-8x37 supports multi-pages application. You may refer to Chapter 3 to setup the multi-page demo – “whmi_05” to see how it work.

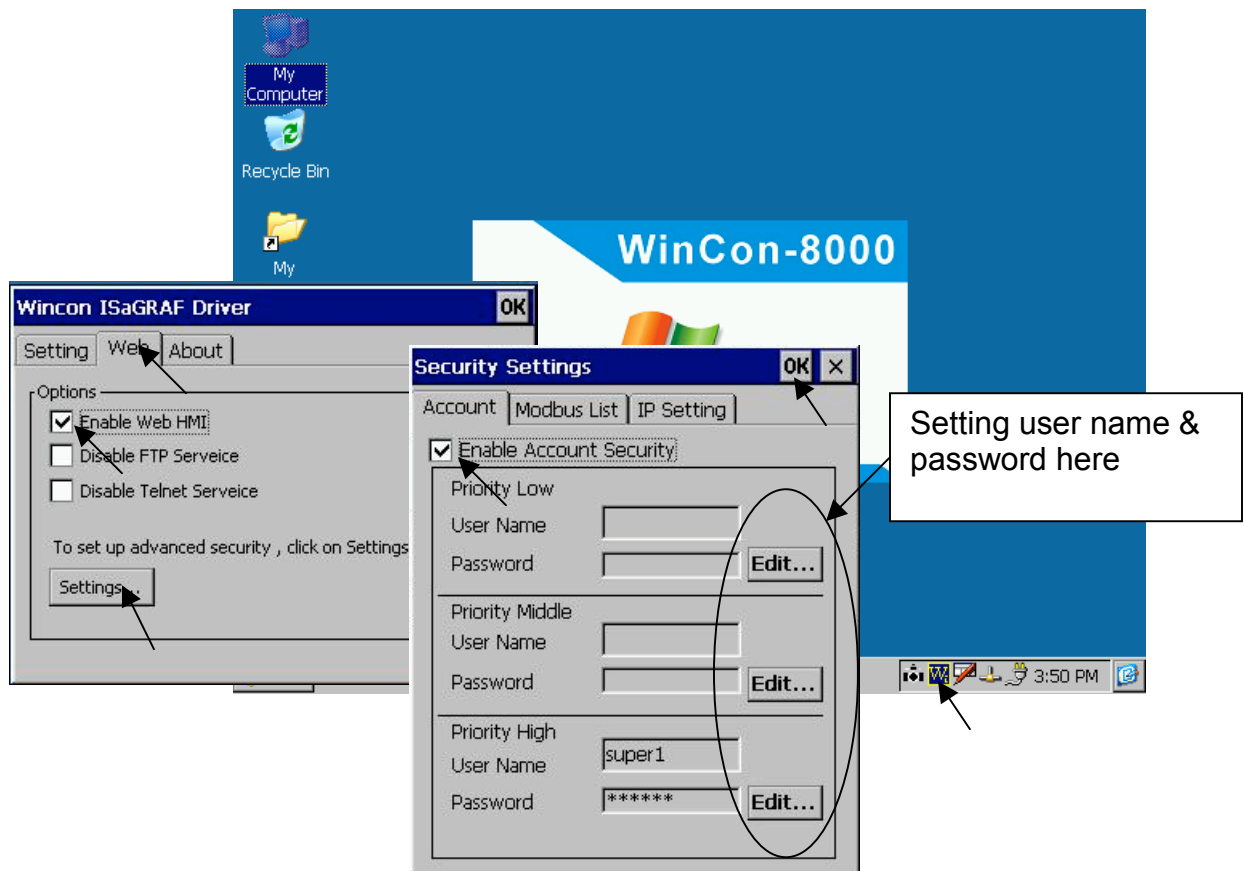
5.5.1: Level 2 And Level 3 Page

The multi-page name can be any valid html file name. For example, “page2.htm”, “kitchen.htm”, “u2-page4.htm” .

If “u2-“ appear in front of the page name, the page will become a Level 2 page. For example, the “u2-Page4.htm” in the “whmi_05” demo. If “u3-“ appear in front of the page name, the page will become a Level 3 page. For example, the “u3-time.htm” in the “whmi_05” demo.

What is a Level2 page ? Only users login with the Middle or High priority can get access to it. To access to the Level3 page, users have to login as a High priority user. The page name without “u2-“ and “u3-“ is identified as Level 1 page. That means any user successfully login can access to it. For example, the “main.htm”

The other rules for multi-pages are almost the same as “main.htm” (section 5.4)



Note: If “Enable Account Security” is not check, any user can easily get access to your Wincon through the Internet Explorer.

5.5.2: Switch One Page To One Another Page

Please take a look at the “menu.htm” of the “whmi_05” demo as below. The “goto_R_page()” function can be used for switching to other page.

```
<!-- top_or_left=0 , scrolling=0 , width=110 , resize=1 -->

<html>
<head>
<title>Title1</title>
<meta http-equiv="Content-Type" content="text/html; charset=big5" >
<SCRIPT LANGUAGE="JavaScript" src="./msg/wincon.js"></SCRIPT>

<SCRIPT LANGUAGE="JavaScript">
function start1()
{
  A_11();
}
function refresh_data()
{
  if(run_at_pc==1) return; // if simulate at the PC, just return

  ...
}
</SCRIPT>
</head>
<body onload="start1()">

<!-- Logout button -->
<form name="form_logout" method="post" action="./login.dll">
  <input style="cursor:hand" name="CMD" type="submit" value="Logout" onClick="return
logout(this.form)">
</form>
<br/>
<br/>

<!-- Goto main.htm -->
<A style="cursor:hand" onClick="goto_R_page('main.htm')">第1頁</A>
<br/>
<br/>

<!-- Goto kitchen.htm -->
<A style="cursor:hand" onClick="goto_R_page('kitchen.htm')">Kitchen</A><br/>
<br/>
<br/>
...
```

“cursor:hand” will display the mouse arrow as a hand when entering the button area

Switch page to “main.htm”

Switch page to “kitchen.htm”

5.6: Web Security

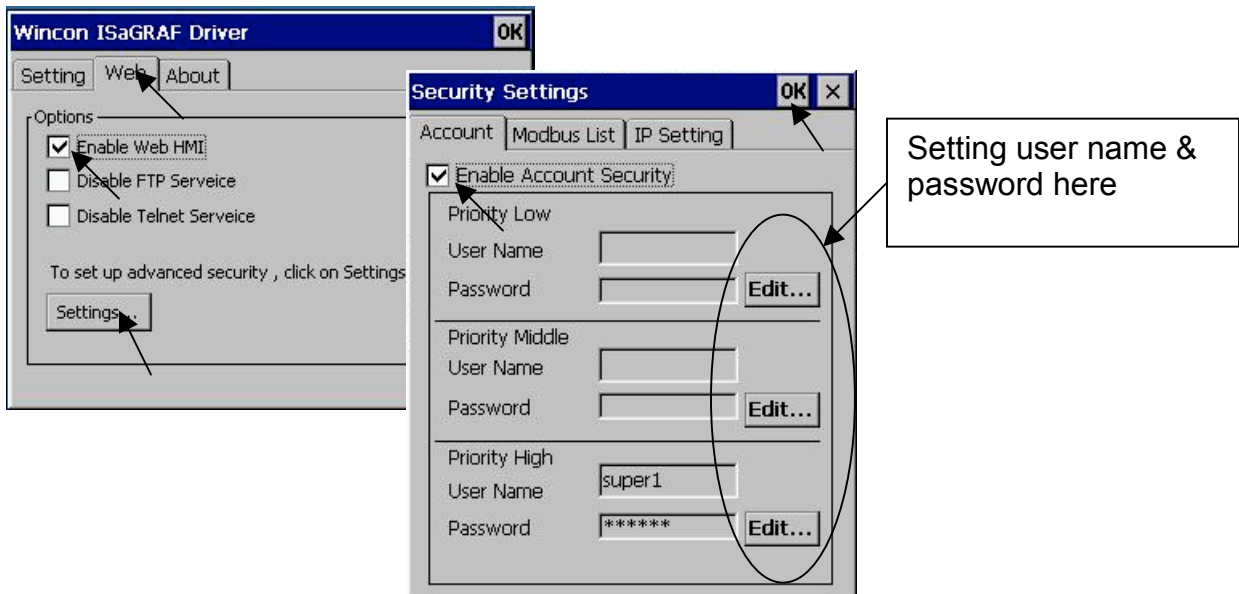
There are some ways user can get access to the Wincon-8037/8337/8737 via its ethernet port.

1. Using Modbus TCP protocol at port No.= 502. (ISaGRAF and other HMI can do this)
2. Using ftp (for example, keyin “ftp://10.0.0.103” on the Internet Explorer)
3. Using telnet (for example, keyin “telnet 10.0.0.103 in the “command” window)
4. Using the Web server (The Web HMI does)

For safety, recommend to disable item 2 and 3 at run time.

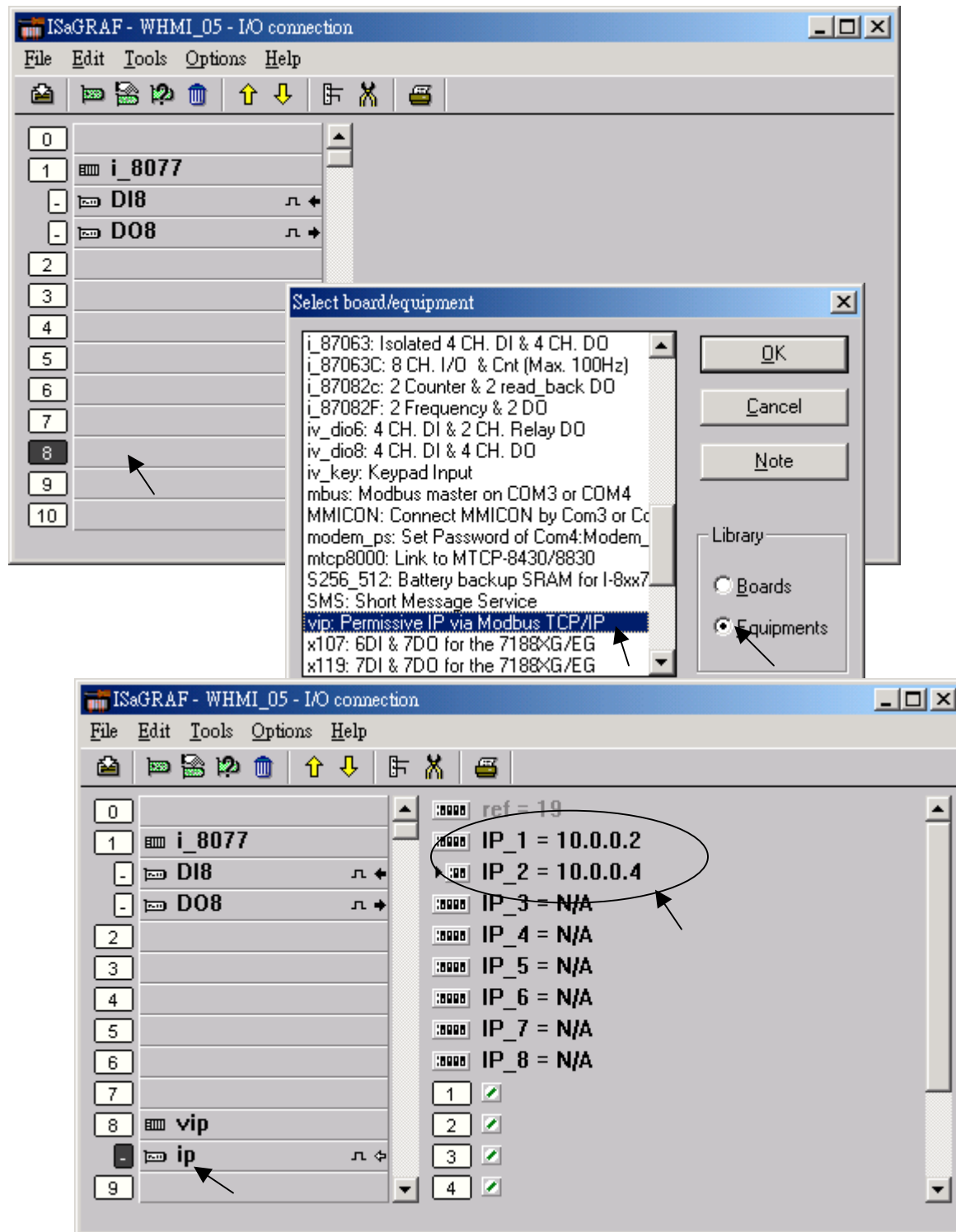


And about item 4, please set proper username & password for the Web HMI.



About item 1, user may set up to 8 IP address for ISaGRAF or other HMI to get access to the W-8x37 via the Modbus TCP/IP protocol as below.

On the IO connection window of ISaGRAF. Please connect “vip” and entering the IP which can get access to the W-8x37 via Modbus TCP/IP protocol. If “vip” is not connected, any remote IP can get access to your W-8x37 via Modbus TCP/IP protocol. If “vip” is connected and No IP is entered (all assigned as “N/A”), No HMI and ISaGRAF can get access to it anymore.



Please re-compile your ISaGRAF project and download it to the controller if you have modified the IO connection.

Chapter 6: VB.net Program Running In Wincon Access To ISaGRAF Variables

This chapter lists the procedure for creating the first demo program by Visual Studio .NET development tool. The ISaGRAF driver of the Wincon-8037/8337/8737 supports "QuickerNet.dll" & "Web HMI" since Rev. of 3.12

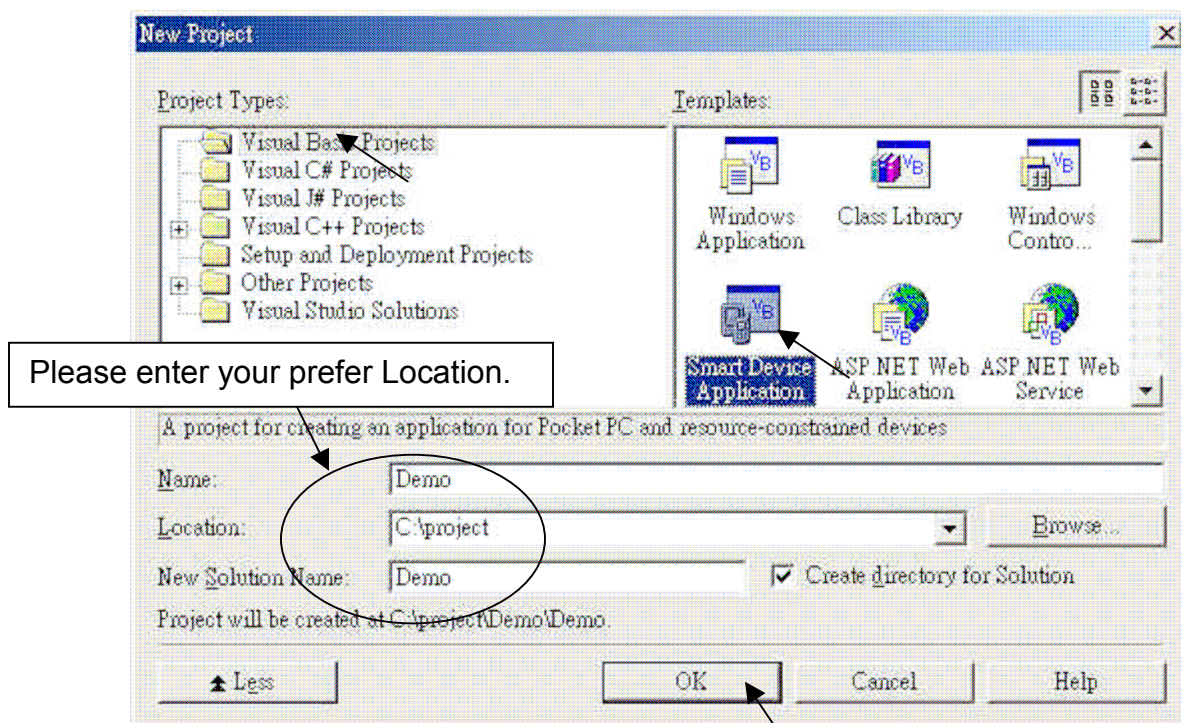
6.1 Create a New SDE Project

Let's start with the traditional "Demo" application, which, while clearly trivial, still illustrates in a nutshell the entire development process. We will assume you have installed both of Visual Studio.NET and the Smart Device Extensions (SDE) development tool. Furthermore, you are trying to develop the user program for applied in Wincon-8000 controller target device. The following steps illustrate in detail for the developing process.

1. In the first, users need to open Microsoft Visual Studio .NET 2003 software. And then in the menu of "**File**", please click the "**New**", and select the project item to open the "**New Project**" dialog.

2. Open the "**New Project**" dialog, and select development tool as following. Here we provide two ways for developing application for Wincon-8000.

(2a.) Select from "**Visual Basic Projects**" and the "**Smart Device Application**" template.



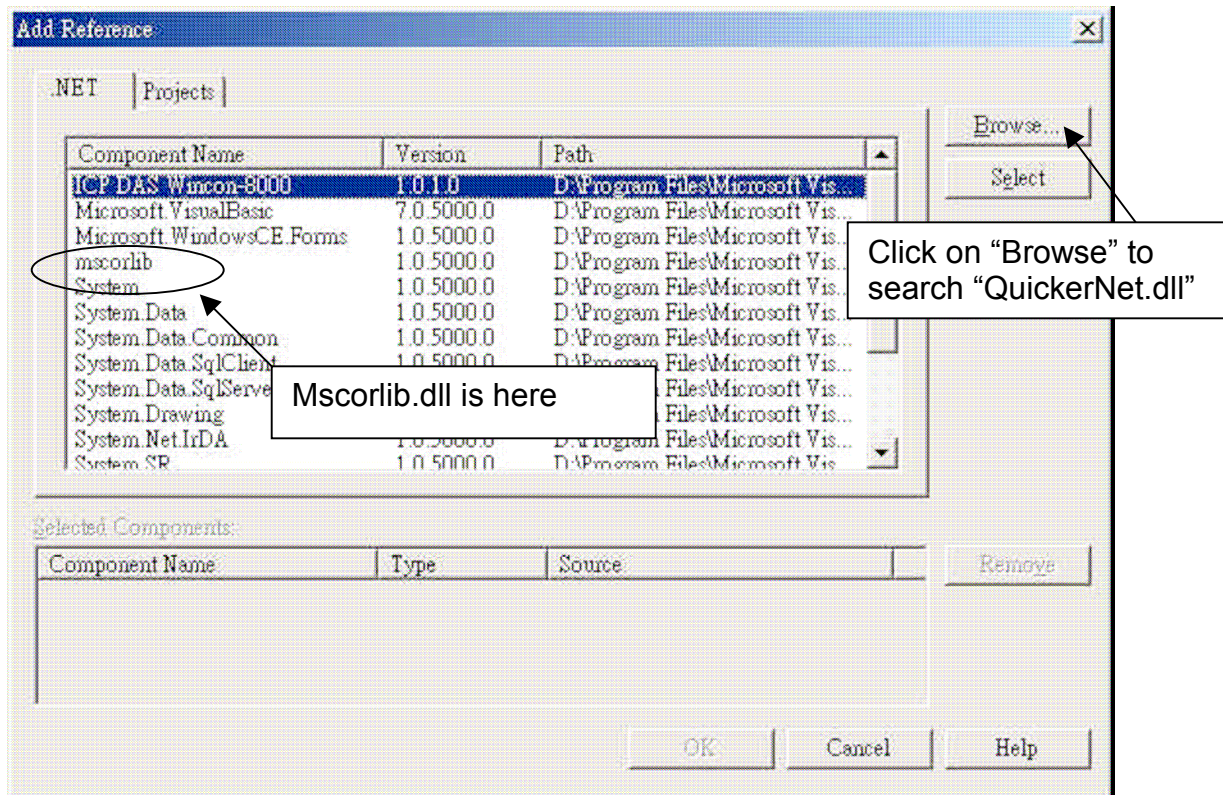
6.2 Add Project Reference for an Application

The “QuickerNet” library contains all modules’ functions. Before you use the “Quicker” keyword in the program design, you must add the “QuickerNet.dll” into the reference list of your application. Please follow the below process to import DLL into the project reference

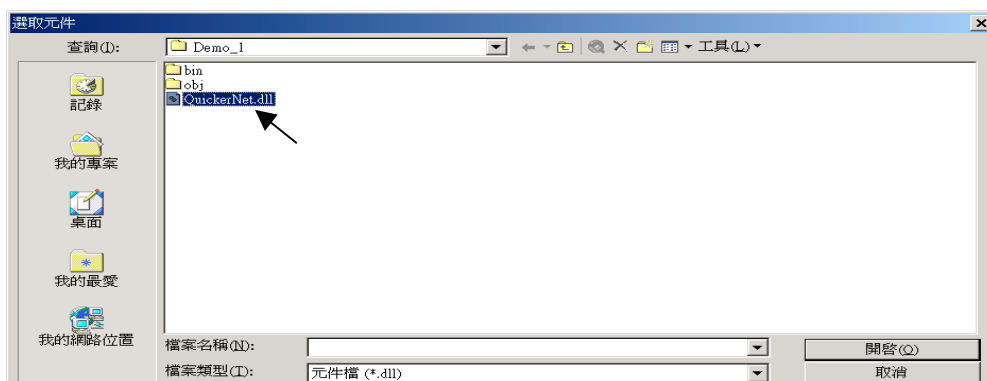
1. Open the “**Add Reference**” dialog by choosing **Project**→**Add Reference**.

There are two libraries we have to add. One is “mscorlib.dll” and the other is “QuickerNet.dll”

Select the “**mscorlib**” in the list box and click the button **Select** (the component “mscorlib” must appear in the Selected Components area)



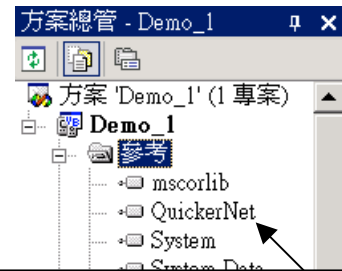
2. Click the “**Browse**” button. Select the “**QuickerNet.dll**” from **Wincon-8xx7 CD**: \napdos\ISaGRAF\Wincon\VB.NET_Demo\VbDmo_01\Demo_1\ subfolder or from your own location.



3. Click **OK** to close the “Add Reference” dialog.

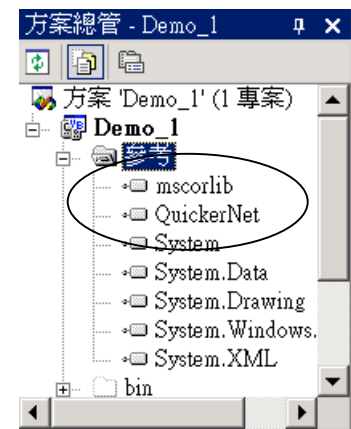
Note: If the “The reference ‘QuickerNet.dll’ may or may not be valid for the active platform.

You should delete old “QuickerNet.dll” reference and then add new “QuickerNet.dll” reference again with care to ensure your application will run correctly” .

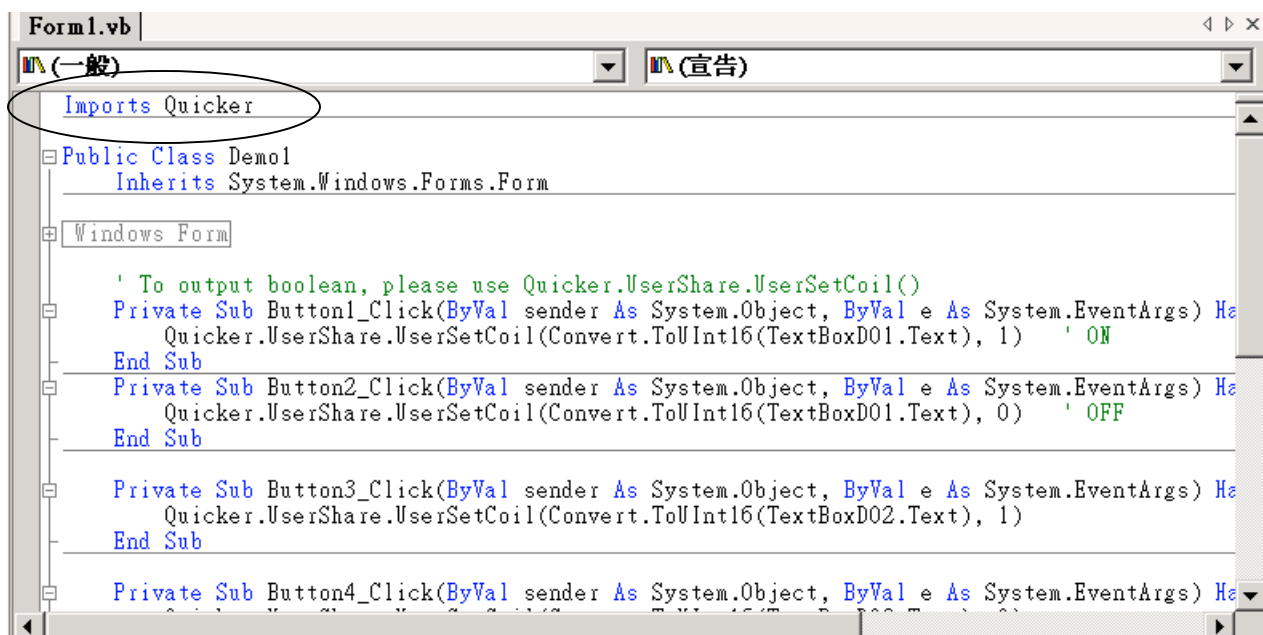


If QuickerNet.dll is not valid, right click on it to delete it and then add it to the reference again.

4. You have completed adding “mscorlib” and “QuickerNet.dll” reference for your application; now expand the “Demo” and “References” folders in the “Solution Explorer” tab to make sure the “QuickerNet.dll” has been added into the project references.



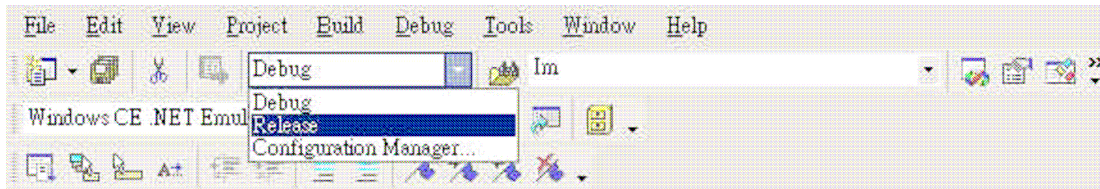
5. Right-click on the “Form1” and select “View Code” from the pop-up. Move cursor to top and insert the “Imports Quicker” in the first statement.



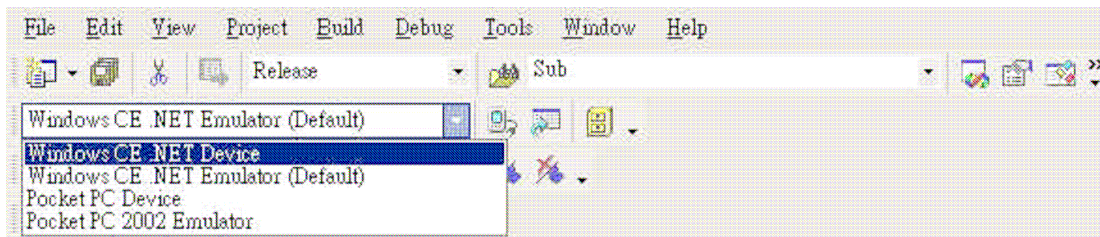
6.3 Building an Application Program

When you have finished writing a demo, you can build an application by the following steps.

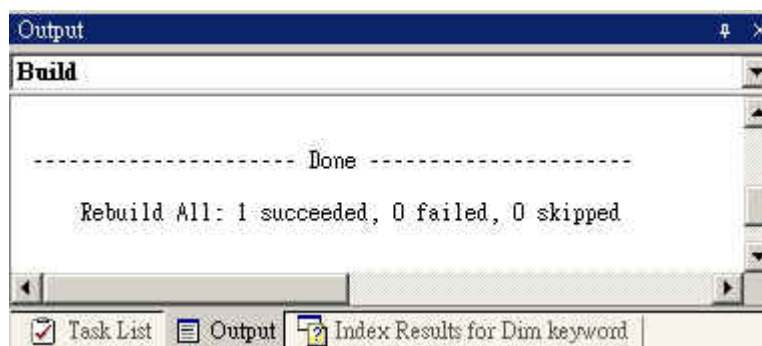
1. In the toolbox menu select “**Release**” from the solution Configurations of Combo box.



2. In the toolbox menu select “**Windows CE .NET Device**” from the Development Device Combo box.



3. On the “**Build**” menu, choose “**Build Demo**”.
4. If you have finished the step 1~4, you will obtain the following message in the Output window, which means the execution application is built. Otherwise, you will get error messages. Therefore, you need to debug the source code and rebuild the application.



5. You can find the Demo.exe file in <Your VB.net Project folder>\bin\Release\demo.exe

6.4 Execute The Application In The Wincon-8xx7 Platform

When you have finished the building process for Demo.exe, You can apply the developed execution file, Demo.exe into the Wincon-8xx7 target controller device.

1. Copy the Demo.exe , Mscorlib.dll , Quicker.DLL and QuickerNet.DLL to your prefer folder of Wincon-8000. (If you are copying file to the \Compact Flash\ISaGRAF\, it already has Mscorlib.dll , Quicker.DLL and QuickerNet.DLL, please do not replace them, just copy your demo.exe into it)
2. Double-click on the Demo.exe file to excute it.

6.5 Examples

There are several VB.NET example demo in the Wincon-8xx7 CD-ROM: \napdos\ISaGRAF\Wincon\VB.NET_Demo". User can copy the files to your hard drive and open the *.sln file to run the project by Microsoft Visual Studio .Net 2003

6.6 QuickerNET.DLL

This section we will focus on the description of the application example of QuickerNET.DLL functions. There are some functions that can be used to R/W data from/to the ISaGRAF softlogic. The functions of QuickerNET.DLL can be clarified as 2 groups as depicted as below:

1. Digital R/W Functions
2. Analog R/W Functions

6.6.1 Digital R/W Functions

■ UserSetCoil

Description:

This function is to set the value to a Boolean variable by Modbus network address.

Syntax:

ScanKernel.UserShare.UserSetCoil (iUserAddress As System.UInt16, iStatus As byte)

Parameter:

iUserAddress : Specify the Modbus Network Address of Variable (1 to 8191)

iStatus : Set the status. For instance, iStatus = 1 for True, iStatus = 0 for False

Return Value:

None

Example:

' Set the output variable of Modbus Network Address "1" to True.
ScanKernel.UserShare.UserSetCoil(Convert.ToUInt16(1), 1)

Demo program :

WINCON8xx7 CD-ROM: \napdos\isagraf\wincon\VB.NET_Demo\vbdmo_01
WINCON8xx7 CD-ROM: \napdos\isagraf\wincon\VB.NET_Demo\vbdmo_02
WINCON8xx7 CD-ROM: \napdos\isagraf\wincon\VB.NET_Demo\vbdmo_03

■ UserGetCoil

Description:

This function is to get the value from a boolean variable by Modbus network address.

Syntax:

ScanKernel.UserShare.UserGetCoil (iUserAddress As System.UInt16, ByRef iStatus As byte)

Parameter:

iUserAddress : Specify the Modbus Network Address of Variable (1 to 8191)
iStatus : Get the variable status , iStatus = 1 for True, iStatus = 0 for False

Return Value:

None

Example:

' Get the variable status of Network Address "1".
Dim iStatus As Byte
ScanKernel.UserShare.UserGetCoil(Convert.ToUInt16(1), iStatus)

Demo program :

WINCON8xx7 CD-ROM: \napdos\isagraf\wincon\VB.NET_Demo\vbdmo_02
WINCON8xx7 CD-ROM: \napdos\isagraf\wincon\VB.NET_Demo\vbdmo_03

6.6.2 Analog R/W Functions

■UserSetReg_short ■UserSetReg_long ■UserSetReg_float

Description:

These functions are to set 16-bit short integer , 32-bit long integer & 32-bit float value to the specified Modbus network address.

Syntax:

```
ScanKernel.UserShare.UserSetReg_Short (ByVal iUserAddress As System.UInt16,  
ByRef iStatus As Integer) As Byte
```

```
ScanKernel.UserShare.UserSetReg_Long (ByVal iUserAddress As System.UInt16,  
ByRef iStatus As Integer) As Byte
```

```
ScanKernel.UserShare.UserSetReg_Float (ByVal iUserAddress As System.UInt16,  
ByRef iStatus As Single) As Byte
```

Parameter:

iUserAddress : Specify the Network Address of Variable (1 to 8191)

iStatus : Set the short or long integer or float value.

Example:

‘ Set a long value “1234567” to the variable of Modbus Network Address “1”.

```
ScanKernel.UserShare.UserSetReg_long(Convert.ToUInt16(1), Convert.ToInt32(1234567) )
```

‘ Set a short value “-1234” to the variable of Modbus Network Address “3”.

```
ScanKernel.UserShare.UserSetReg_short(Convert.ToUInt16(3), Convert.ToInt16(-1234) )
```

‘ Set a float value “2.174” to the variable of Modbus Network Address “4”.

```
ScanKernel.UserShare.UserSetReg_float(Convert.ToUInt16(4), Convert.ToSingle(2.174) )
```

Demo program :

Please refer to WINCON8xx7 CD-ROM

\\napdos\isagraf\wincon\VB.NET_Demo\vbdmo_05 & vbdmo_06 for R/W Analog Input / Output (16-bit short)

2. \\napdos\isagraf\wincon\VB.NET_Demo\vbdmo_07 for R/W long integer value (32-bit) , float value (32-bit) & Timer (32-bit)

Note:

The long integer & timer & float variable's Network Address No. must occupy 2 No. in the ISaGRAF project (refer to section 4.2 of “User's Manual of ISaGRAF Embedded Controllers” or in the CD-ROM: \\napdos\isagraf\wincon\english_manu\” User_Manual_I_8xx7.pdf”)

■UserGetReg_short ■UserGetReg_long ■UserGetReg_float

Description:

These functions are to get 16-bit short integer , 32-bit long integer & 32-bit float value from the specified Modbus network address.

Syntax:

```
ScanKernel.UserShare. UserGetReg_Short (ByVal iUserAddress As System.UInt16,  
ByRef iStatus As Integer) As Byte
```

```
ScanKernel.UserShare. UserGetReg_Long (ByVal iUserAddress As System.UInt16,  
ByRef iStatus As Integer) As Byte
```

```
ScanKernel.UserShare. UserGetReg_Float (ByVal iUserAddress As System.UInt16,  
ByRef iStatus As Single) As Byte
```

Parameter:

iUserAddress : Specify the Network Address of Variable (1 to 8191)

iStatus : Get the short or long integer or float value.

Example:

```
Dim float_val As Single
```

```
Dim short_val As Int16
```

```
Dim long_val As Int32
```

‘ Get float value of the variable of Modbus Network Address “7”.

```
ScanKernel.UserShare.UserGetReg_float(Convert.ToUInt16(7), float_val)
```

‘ Get long value of the variable of Modbus Network Address “9”.

```
ScanKernel.UserShare.UserGetReg_long(Convert.ToUInt16(9), long_val)
```

‘ Get short value of the variable of Modbus Network Address “11”.

```
ScanKernel.UserShare.UserGetReg_short(Convert.ToUInt16(11), short_val)
```

Demo program :

Please refer to WINCON8xx7 CD-ROM

1. \napdos\isagraf\wincon\VB.NET_Demo\vbdmo_05 & vbdmo_06 for R/W Analog Input / Output (16-bit short)

2. \napdos\isagraf\wincon\VB.NET_Demo\vbdmo_07 for R/W long integer value (32-bit) , float value (32-bit) & Timer (32-bit)

Note:

The long integer & timer & float variable’s Network Address No. must occupy 2 No. in the ISaGRAF project (refer to section 4.2 of “User’s Manual of ISaGRAF Embedded Controllers” or in the CD-ROM:\napdos\isagraf\wincon\english_manu\” User_Manual_I_8xx7.pdf”)

Chapter 7: EVC++ Program Running In Wincon Access To ISaGRAF Variables

User can write his EVC++ 4.0 application to access to the ISaGRAF variables running at the same Wincon by using the below functions for Read/Write boolean, word, long and float value.

The include file and library at design time are "WinConAgent.h" and "Quicker.lib". (Wincon-8x37 CD-ROM:\napdos\isagraf\wincon\EVC++_LIB\). The DLL at run time is the "Quicker.dll" which is in Wincon - \Compact Flash\ISaGRAF\ (Please copy and run your EVC++ application at Wincon - \Compact Flash\ISaGRAF\)

Set boolean value:

```
unsigned char UserSetCoil(unsigned short iUserAddress, unsigned char iStatus);
```

iUserAddress: 1 to 8191 (Variable's network address in ISaGRAF project)

iStatus: 0: set boolean to False, 1: set boolean to True

for ex. UserSetCoil(100 , 1) // set boolean at network addr 100 as True

Set word or float or long value:

```
unsigned char UserSetReg(unsigned short iUserAddress, long *iStatus, unsigned char iDType);
```

iUserAddress: 1 to 8191 (Variable's network address in ISaGRAF project)

iStatus: A pointer to a long type, which stores the data to set

iDType 0: type is word

1: data type is float

2: data type is long (use long for Timer value in ISaGRAF, unit is ms)

for ex.

```
float float_val;
```

```
long word_val, long_val;
```

```
long *temp_val;
```

```
// set word_val (-32768 to +32767) to ISaGRAF variable with network address 1
```

```
word_val = -20000 ;
```

```
temp_val = (long *)&word_val;
```

```
UserSetReg(1 , temp_val, 0);
```

```
// set float_val to ISaGRAF variable with network address 2
```

```
float_val = 1.2345 ;
```

```
temp_val = (long *)&float_val;
```

```
UserSetReg(2 , temp_val, 1);
```

```
// set long_val to ISaGRAF variable with network address 4
```

```
long_val = 12345678 ;
```

```
temp_val = (long *)&long_val;
```

```
UserSetReg(4 , temp_val, 2);
```


Get boolean value:

```
unsigned char UserGetCoil(unsigned short iUserAddress, unsigned char *iStatus);
```

iUserAddress: 1 to 8191 (Variable's network address in ISaGRAF project)
iStatus: 0: boolean is False, 1: boolean is True

for ex.

```
unsigned char bVal;  
UserGetCoil(5, &bVal) // get boolean value at network addr 5
```

Get word or float or long value:

```
unsigned char UserGetReg(unsigned short iUserAddress, long *iStatus, unsigned char iDType);
```

iUserAddress: 1 to 8191 (Variable's network address in ISaGRAF project)
iStatus: A pointer to a long type, which stores the data returned
iDType: 0: type is word
1: data type is float
2: data type is long (use long for Timer value in ISaGRAF, unit is ms)

for ex.

```
float float_val;  
long word_val, long_val;  
long ret_val;  
  
// get word_val (-32768 to +32767) of ISaGRAF variable with network address 10  
UserGetReg(10, &ret_val, 0);  
if ( ret_val >= 0 && ret_val <= 32767 ) word_val = ret_val;  
else word_val = ret_val | 0xFFFF0000;  
  
// get float of ISaGRAF variable with network address 11  
UserGetReg(11, &ret_val, 1);  
float_val = *(float *) (&ret_val);  
  
// get long of ISaGRAF variable with network address 13  
UserGetReg(13, &ret_val, 2);  
long_val = ret_val;
```

Note:

The long integer & timer & float variable's Network Address No. must occupy 2 No. in the ISaGRAF project (refer to section 4.2 of "User's Manual of ISaGRAF Embedded Controllers" or in the CD-ROM:\napdos\isagraf\wincon\english_manu\ "User_Manual_I_8xx7.pdf")

Chapter 8: InduSoft Project Running In Wincon Access To ISaGRAF Variables

Please refer to CD-ROM: \napdos\isagraf\wincon\english_manu\ "user_manual_i_8xx7.pdf" for detailed ISaGRAF English User's Manual.

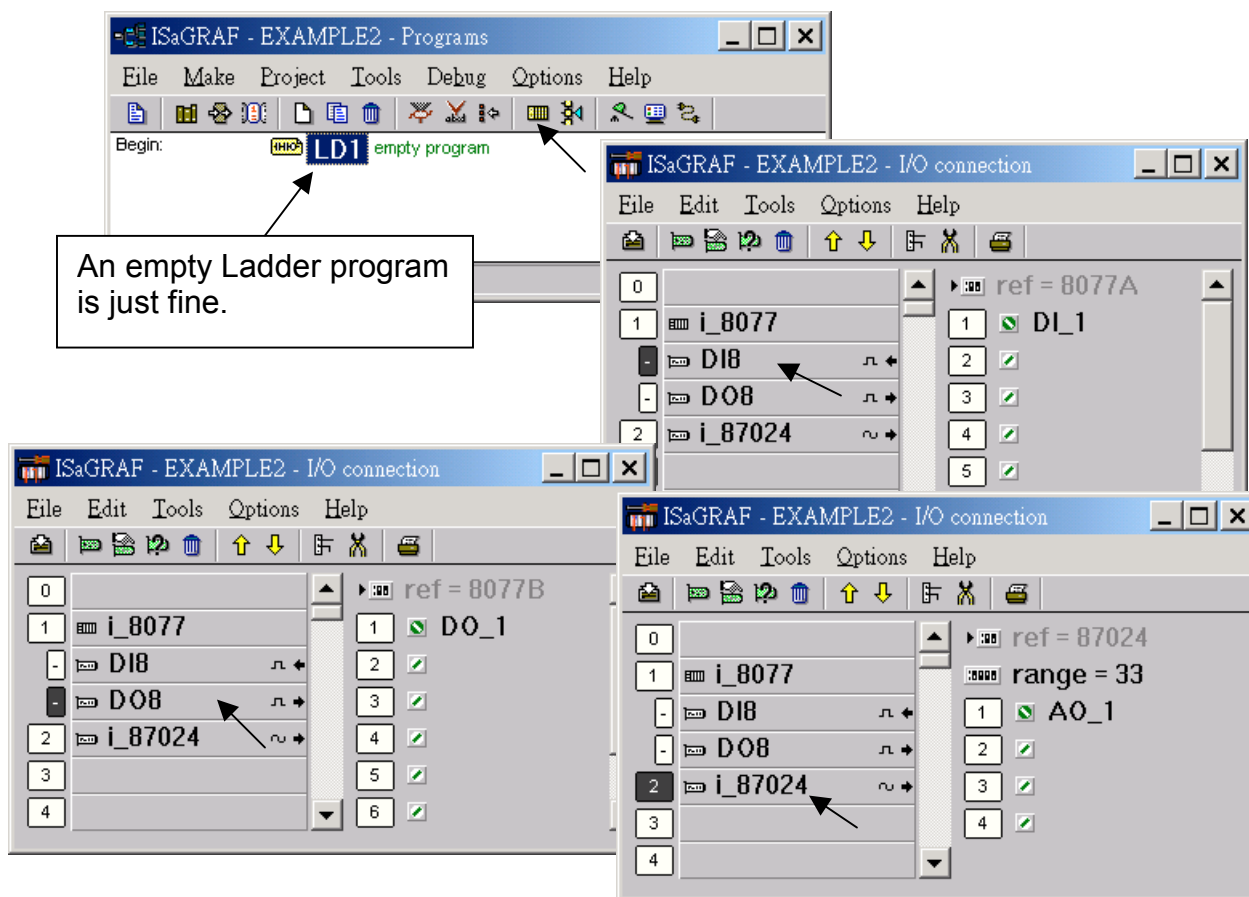
Wincon-8036/8336/8736 supports InduSoft and ISaGRAF control logic running in the same controller.

A simple example to run InduSoft & ISaGRAF logic in the same controller:

Step 1: Create a new ISaGRAF project as below.

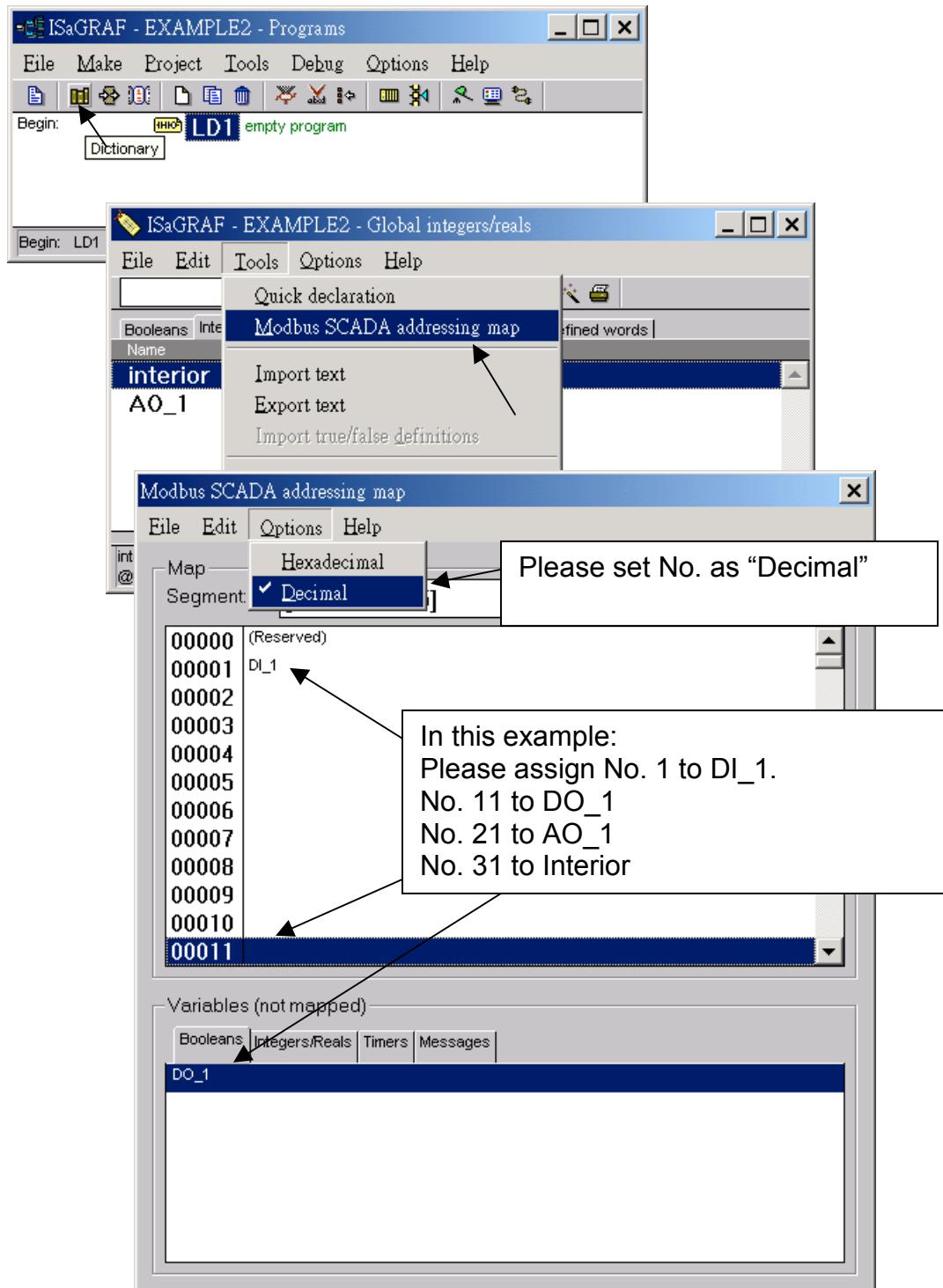
This demo uses a DIO module I-8077 in slot 1 of W-8336, and an AO module I-87024 in slot 2 and one internal variable defined as follow.

Variable Type	Name	Network Address	Comment	Attributes
Boolean	DI_1	1	8077 DI channel 1	Input
Boolean	DO_1	11	8077 DO channel 1	Output
Integers	AO_1	21	87024 AO channel 1	Output
Integers	Interior	31	Internal variable	Internal



If you are not familiar with ISaGRAF, please refer to section 4.1 to 4.3.

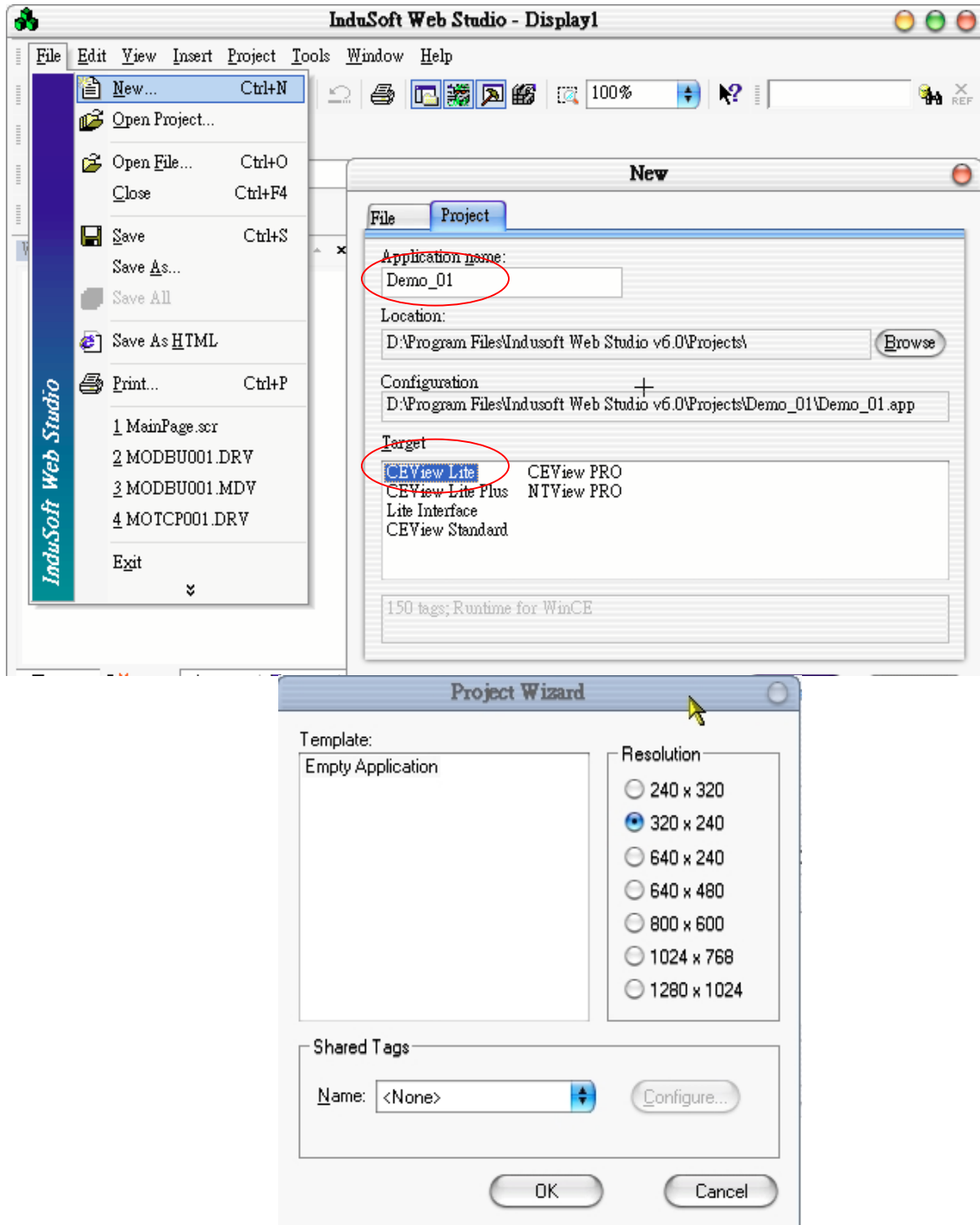
The ISaGRAF variables to be exchanged with Indusoft must be declared with a Modbus “Network Address” as below.



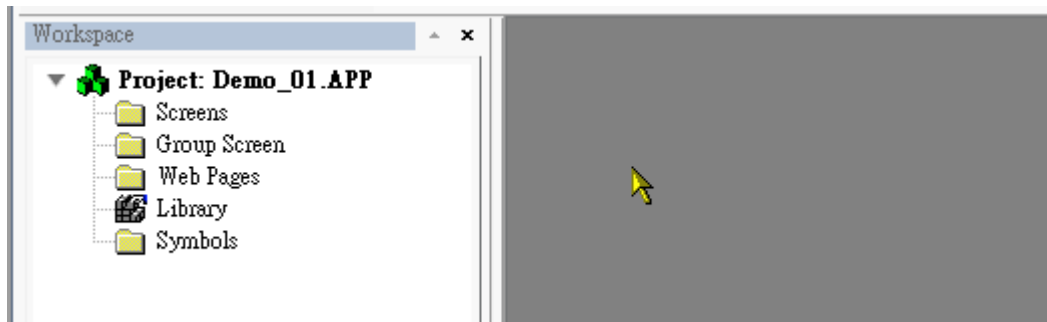
Please save & compile the ISaGRAF example project & then download to the W-8336. If you are not familiar with it, please refer to Section 4.1 to 4.3

Step 2: Create an Indusoft project.

Select “File -> New” from the “InduSoft Web Studio” main menu, the “New” window will appear and click on “Project” tab. Then type in the name for the new user’s project in the “Application name” and select “CEView Lite” in the “Target”. Press “OK” and the “Project Wizard” window will appear. Select “Empty Application” on the “Template”, “320 x 240” on the “Resolution” and “None” on the “Shared Tags”.

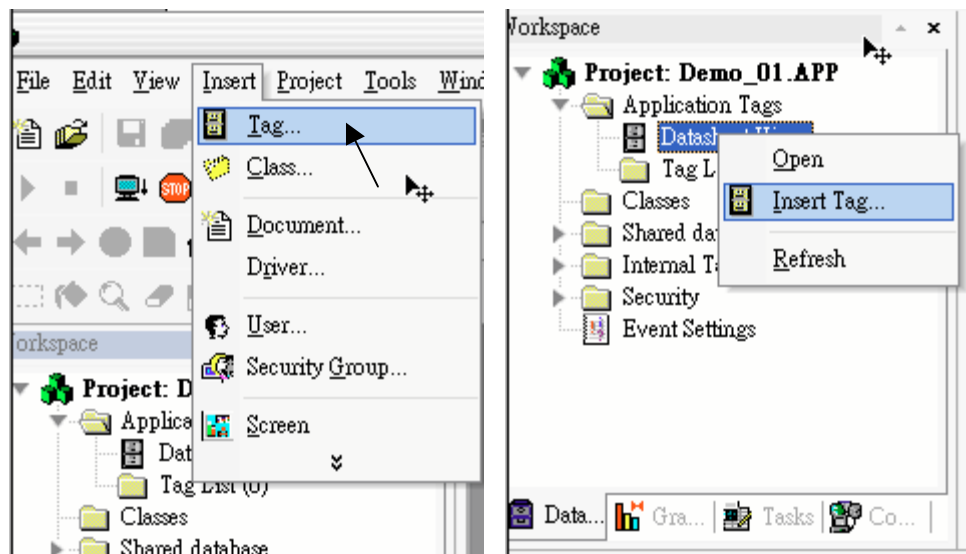


Now you could see the new project shown on the “Workspace” window as below.

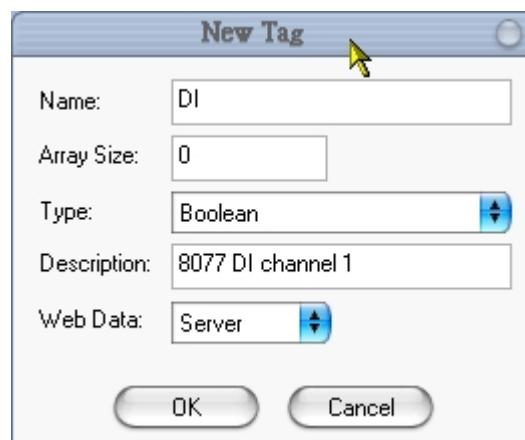


Define application tags

Select “Insert -> Tag” on the main menu bar or click on the right button of the mouse in the “Database” tab of the “Workspace”.



The “New Tag” window will show as below.

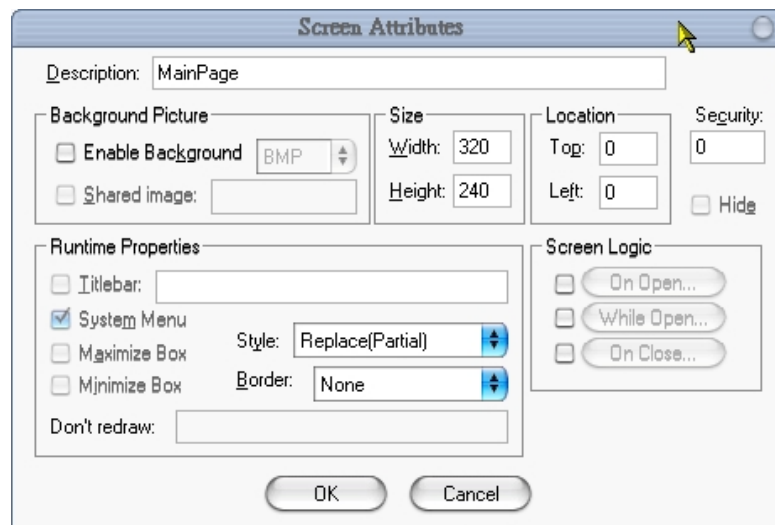
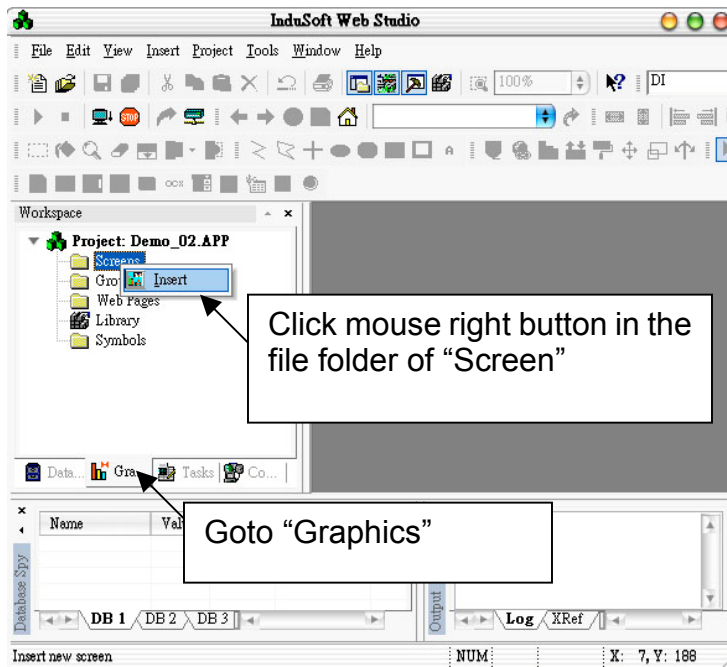


This demo uses a DIO module I-8077, an AO module I-87024 and one internal variable defined as follow. Please create these tags one by one.

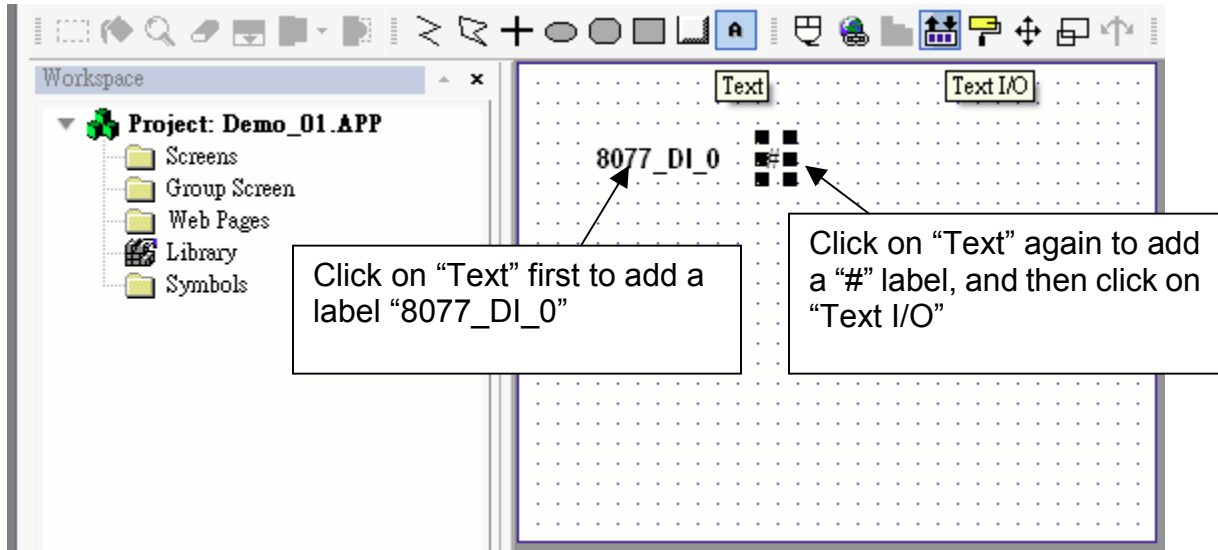
Type	Name	Array Size	Description	Web Data
Boolean	DI	0	8077 DI channel 1	Input
Boolean	DO	0	8077 DO channel 1	Output
Integers	AO	0	87024 AO channel 1	Output
Integers	Interior	0	Internal variable	Internal

Create main screen

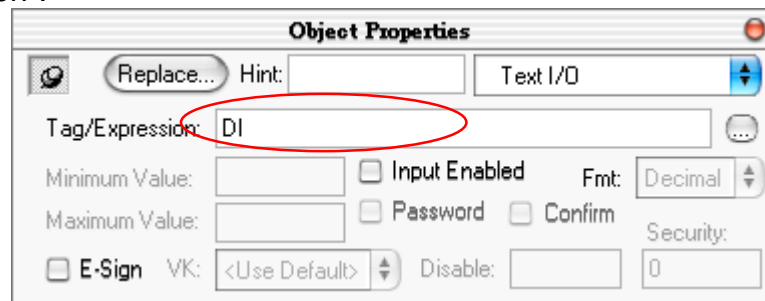
Click mouse right button in the file folder of “Screen” of the “Graphics” tab in the “Workspace” then the “Screen Attributes” window appears. Setting screen attributes such as “Size”, “Location”, “Runtime Properties” and “Background Picture” then press “OK” to edit screen.



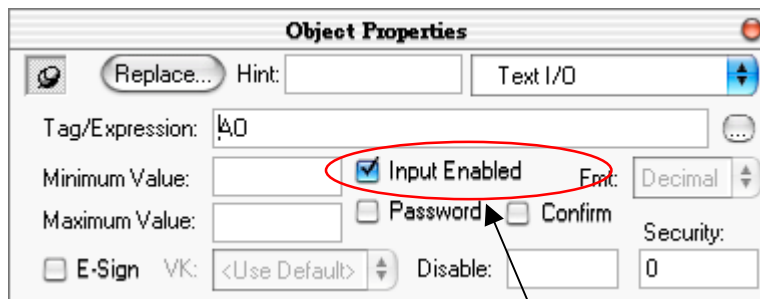
Select “Text” icon, then click on the main screen where want to establish a text and type “8077_DI_0”. And select “Text” again following the previous text and type “#” then select “Text I/O”



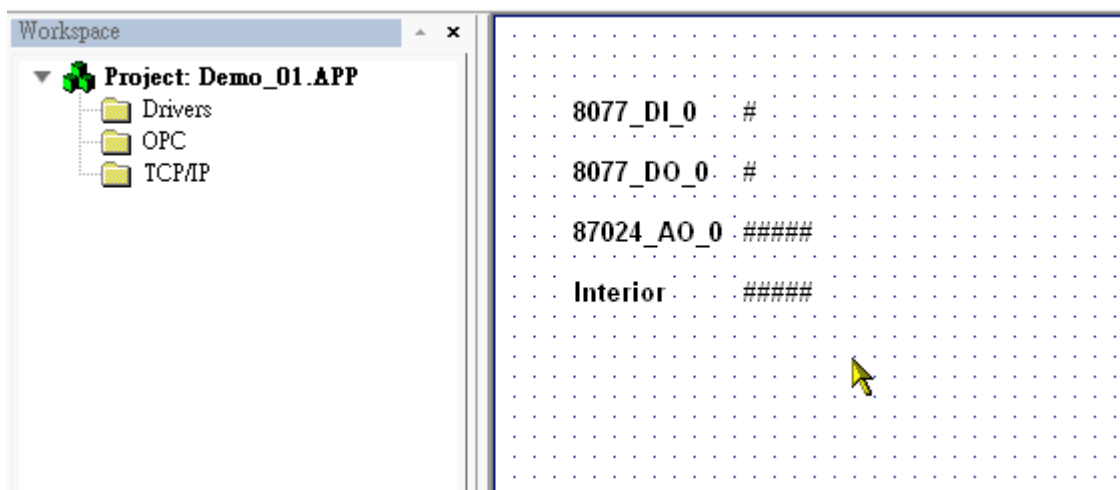
Double click the “#” object and the “Object Properties” window will show as below then type DI in the “Tag/Expression”.



Repeat former method to create other objects and click “Save” icon on the main menu to save this main screen page as “MainPage.scr”. Note: For the Output object, as 87024_AO_0, the “Input Enabled” of the “Text I/O” should be checked as below.

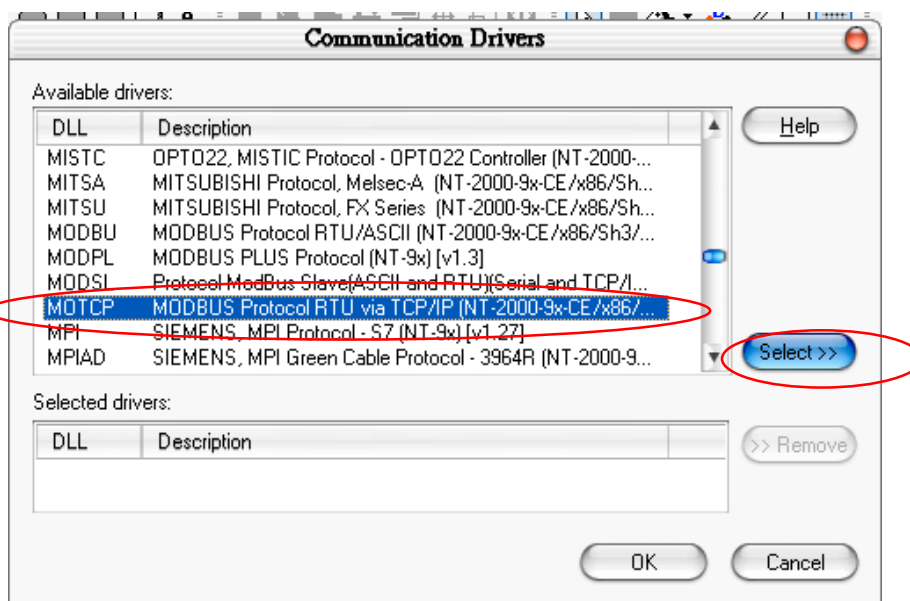
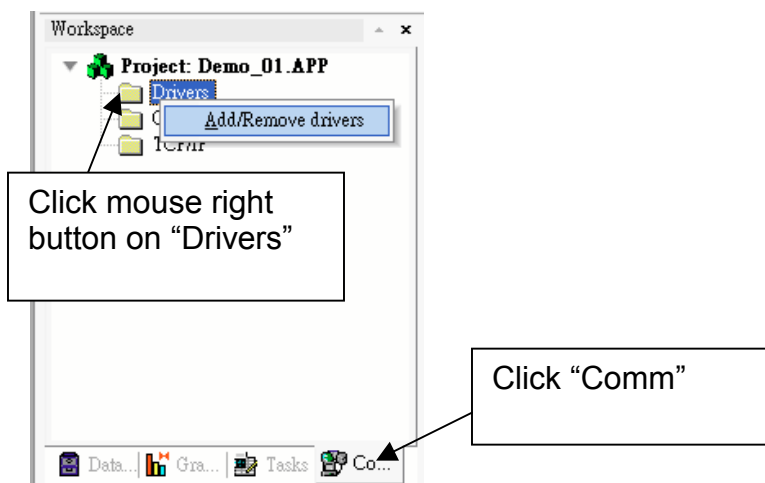


Check on it means user may input its value at run time.

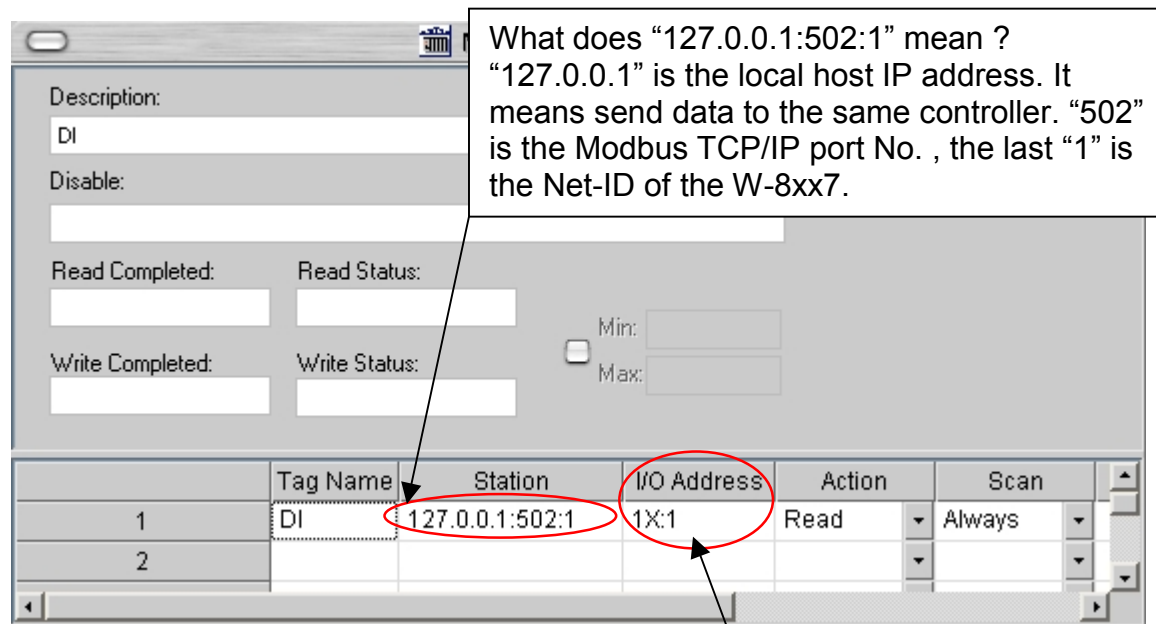
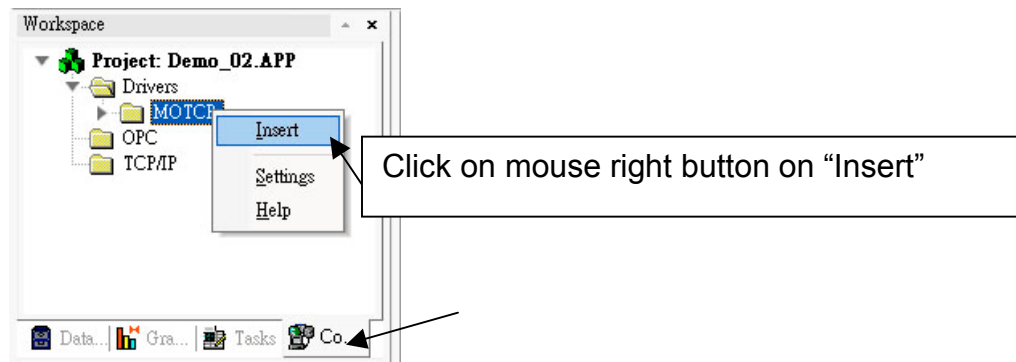


Create Modbus TCP workspace

Click "Comm" in the "Workspace" and in the folder of "Drivers" click right mouse button and select "Add/Remove drivers" to open "Communication Drivers" window as below.



Click “MOTCP” driver then click “Select” and click “OK” to close this window. Expanding file folder of “Drivers” and it will show a file folder named “MOTCP”. Click right mouse button and select “Insert” to add a workspace of Modbus TCP. When a “MOTCP001.DRV” window appears, fill in following data as corresponding field.



1X:1 is for reading “Boolean” data, address 1
 0X:2 is for writing “Boolean” data”, address 2
 3X:5 is for reading short “integer” data (16-bit integer, Word: -32768 to +32767), address 5
 4X:6 is for writing short “integer” data (16-bit integer , Word: -32768 to +32767), address 6
 DW:10: is for reading & writing long “integer” (32-bit integer, Double Word), address 10
 FP:12: is for reading & writing floating point data (32-bit REAL), address 12
 For more details, please refer form as below

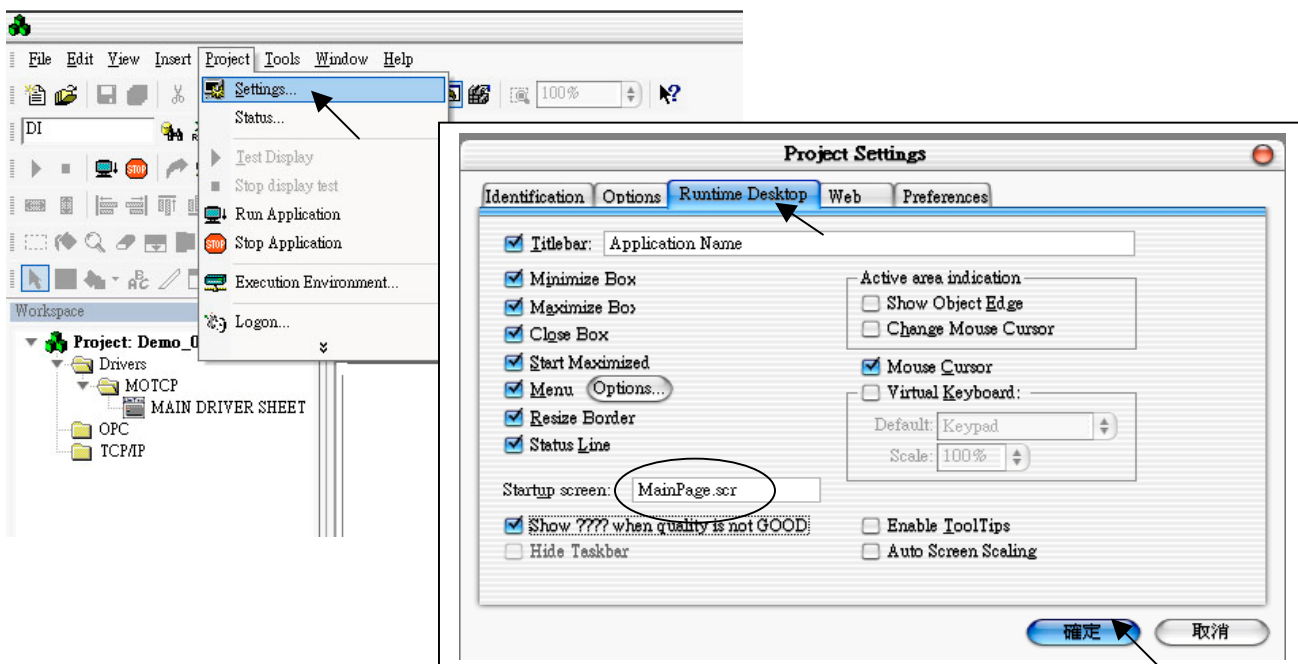
Data Type	Sample Syntax	Valid Range of Initial Addresses per Worksheet	Comments
0X	0X:1	Varies according to the equipment	Coil Status: Read and write events using Modbus instructions 01, 05, and 15
1X	1X:5	Varies according to the equipment	Input Status: Read events using Modbus instructions 02
3X	3X:4	Varies according to the equipment	Input Register: Read events using Modbus instruction 04
4X	4X:5	Varies according to the equipment	Holding Register: Read and write events using Modbus instructions 03, 06, 16
FP	FP:1	Varies according to the equipment	Floating-point value (Holding Register): Read and write float-point values using two consecutive Holding Registers.
DW	DW:2	Varies according to the equipment	32-bit Integer value (Holding Register): Read and write 32-bit integer values using two consecutive Holding Registers.

DRV Name	MOTCP001.DRV	MOTCP002.DRV	MOTCP003.DRV	MOTCP004.DRV
Description	DI	DO	AO	Internal
Station	127.0.0.1:502:1			
IO Address	1X:1	0X:11	4X:21	3X:31
Tag Name	DI	DO	AO	Interior
Action	Read	Write	Write	Read+Write

When finished all setting, press “Ctrl + F4” to close all inside windows and save all files.

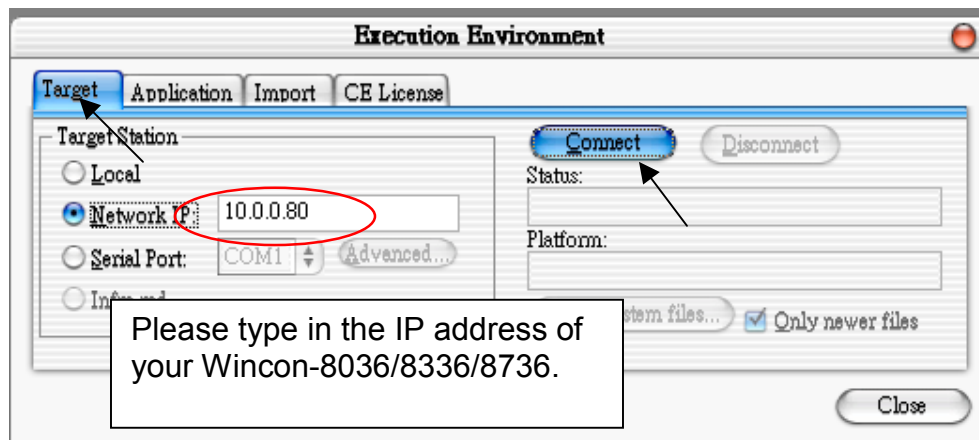
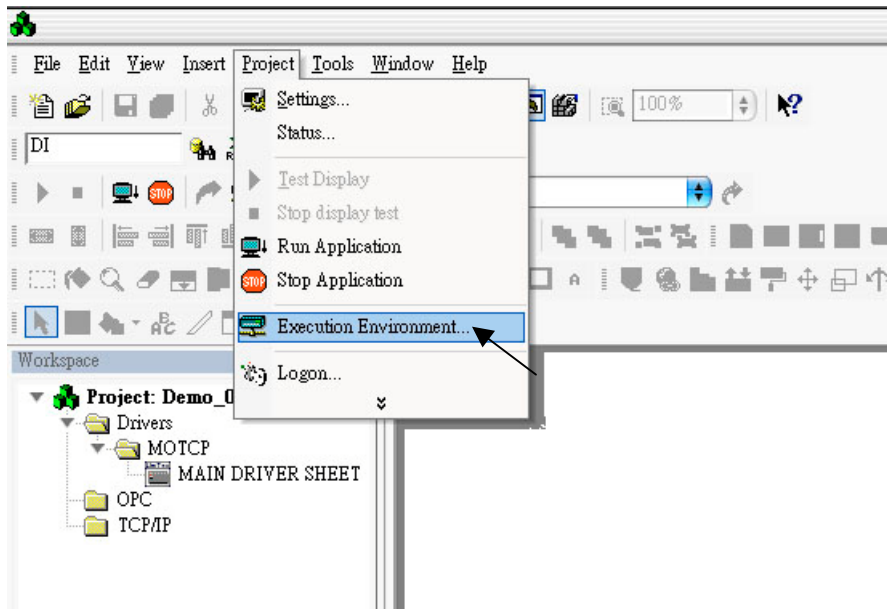
Project Setting

Select “Project -> Settings” to open “Project Settings” window .Click the tab of “Runtime Desktop”. In the “Startup screen” edit box, fill in “MainPage.scr” then click “OK” to close this window.

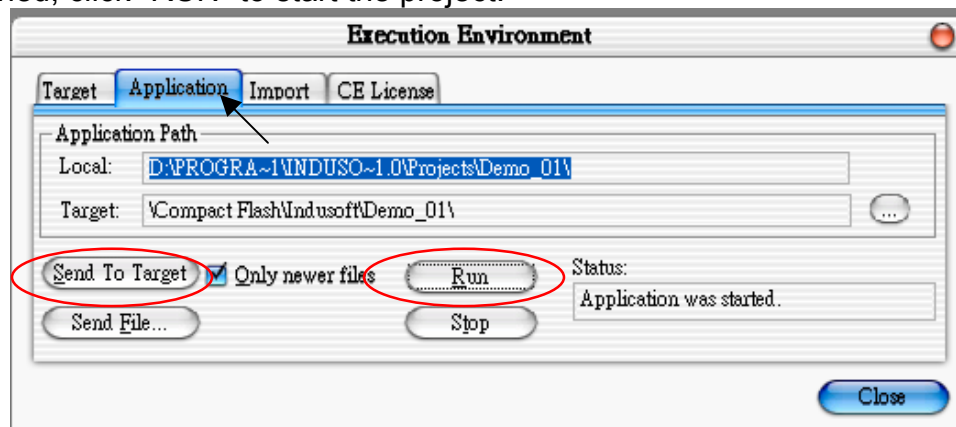


Download and run the project

Select “Project -> Execution Environment” to open “Execution Environment” window. On the Target tab, select “Network IP” then type W-8x36’s correct IP address and click “Connect”.



If connection is fine, click on the tab of “Application” then click “Send to Target”. When download finished, click “RUN” to start the project.



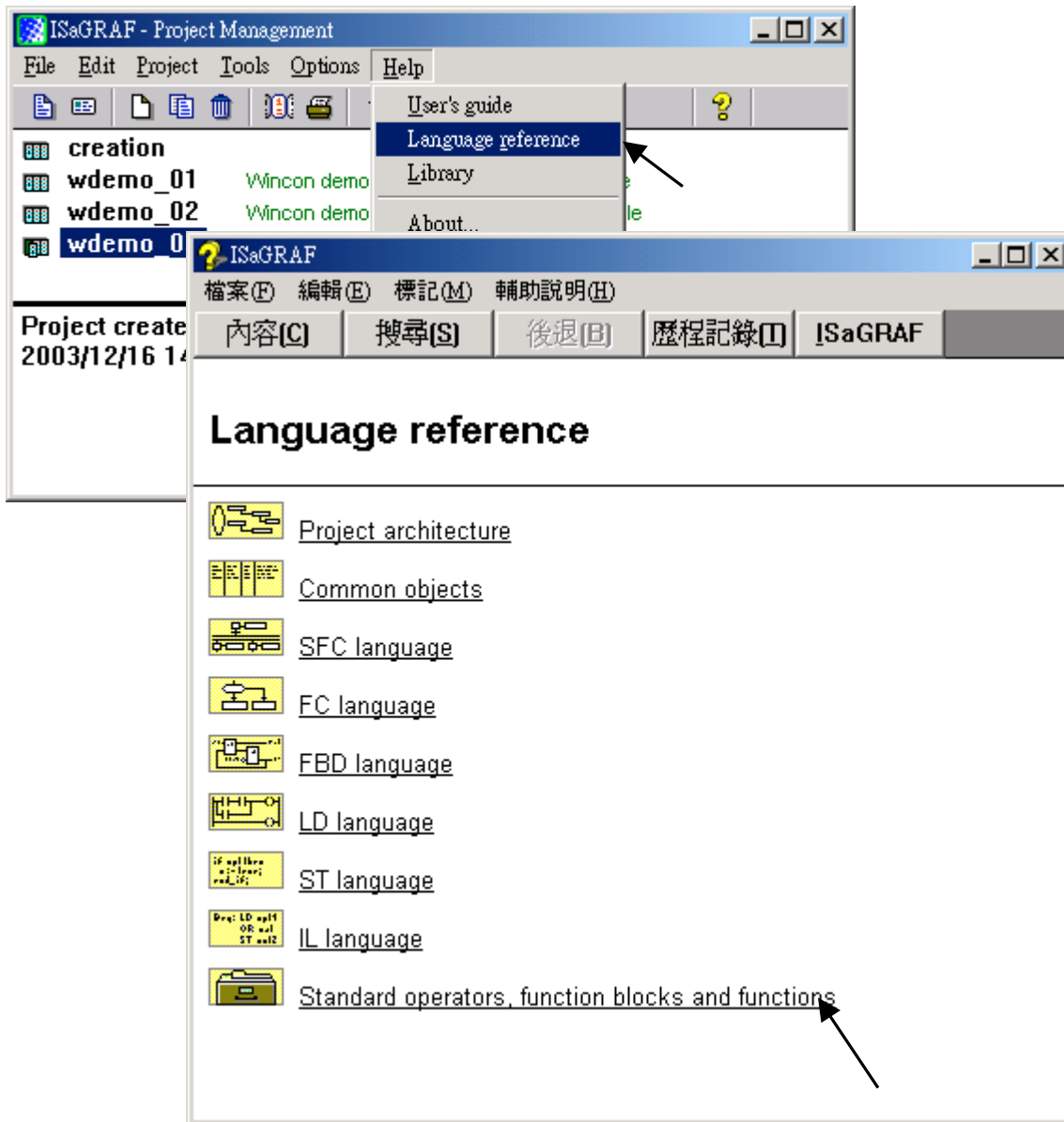
Chapter 9: ISaGRAF Example Program

Please refer to CD-ROM: \napdos\isagraf\wincon\english_manu\ "user_manual_i_8xx7.pdf" for detailed English User's Manual.

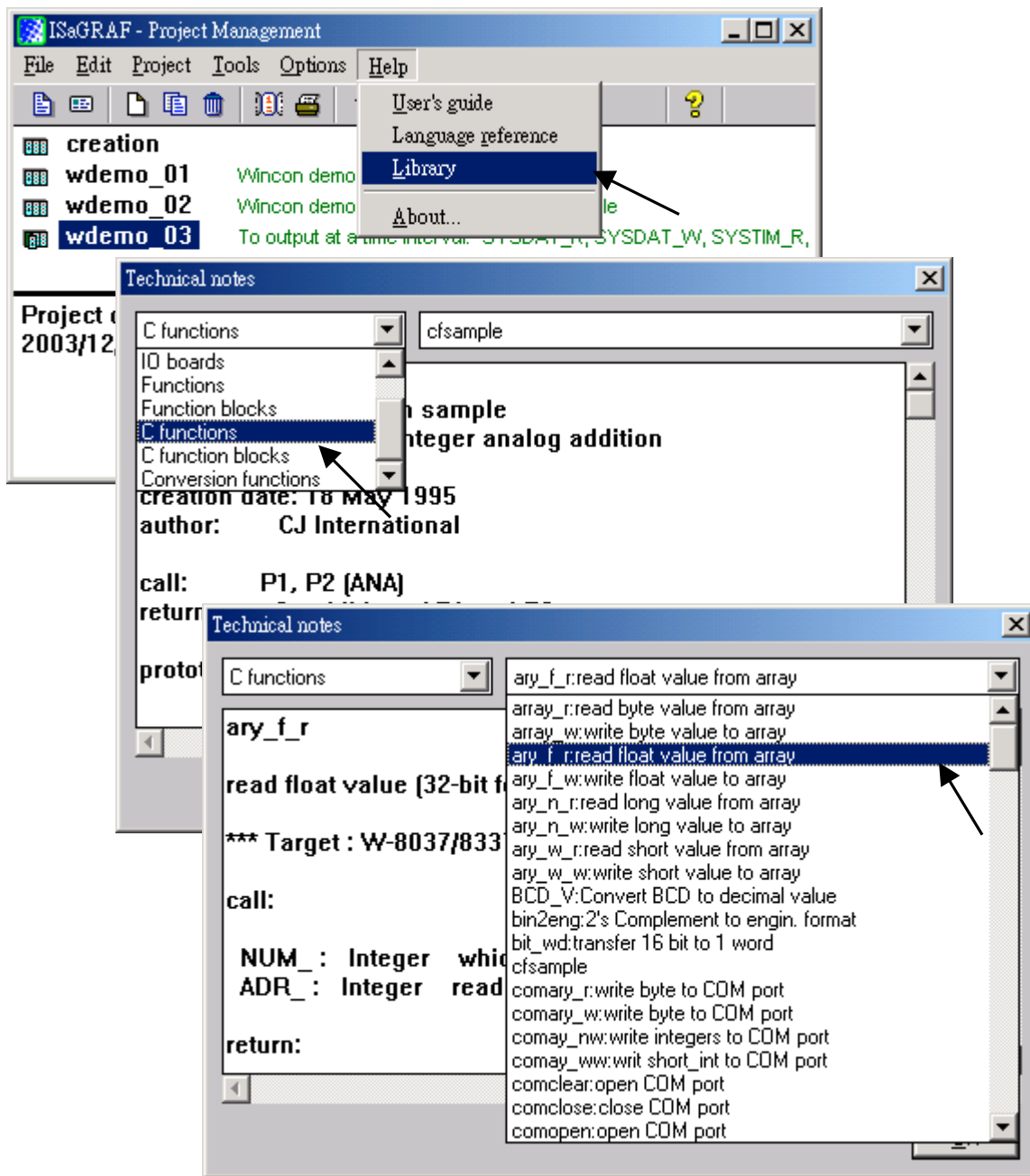
9.1: Get On-Line help

If you have question, you may email to service@icpdas.com.

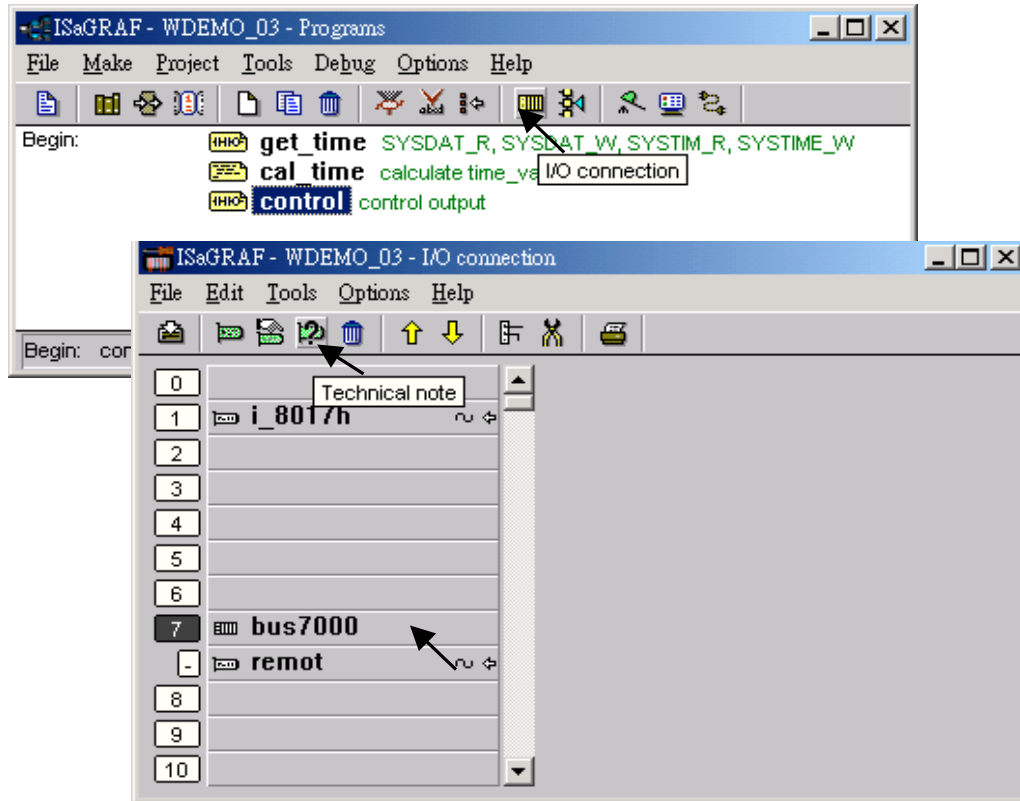
On-line help of ISaGRAF standard functions & function blocks:



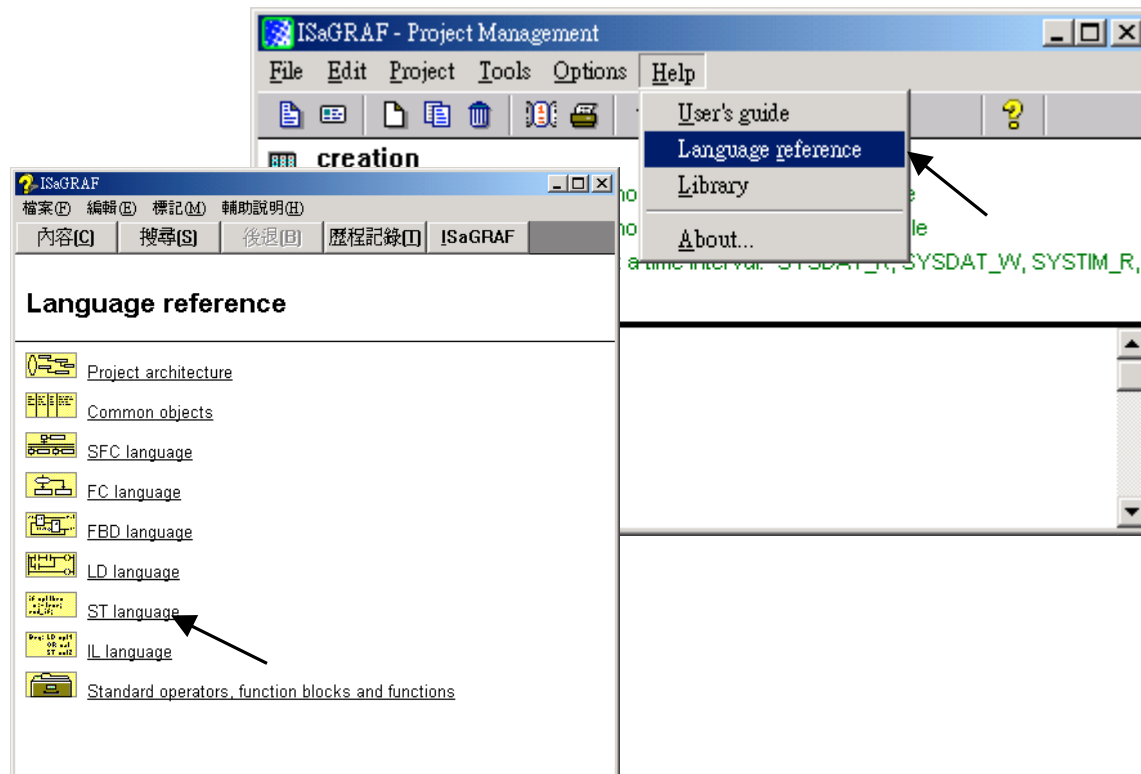
On-line help of ICP DAS add-on functions & function blocks:



On-line help of ICP DAS add-on I/O boards & I/O complex equipments:



On-line help of ISaGRAF languages:



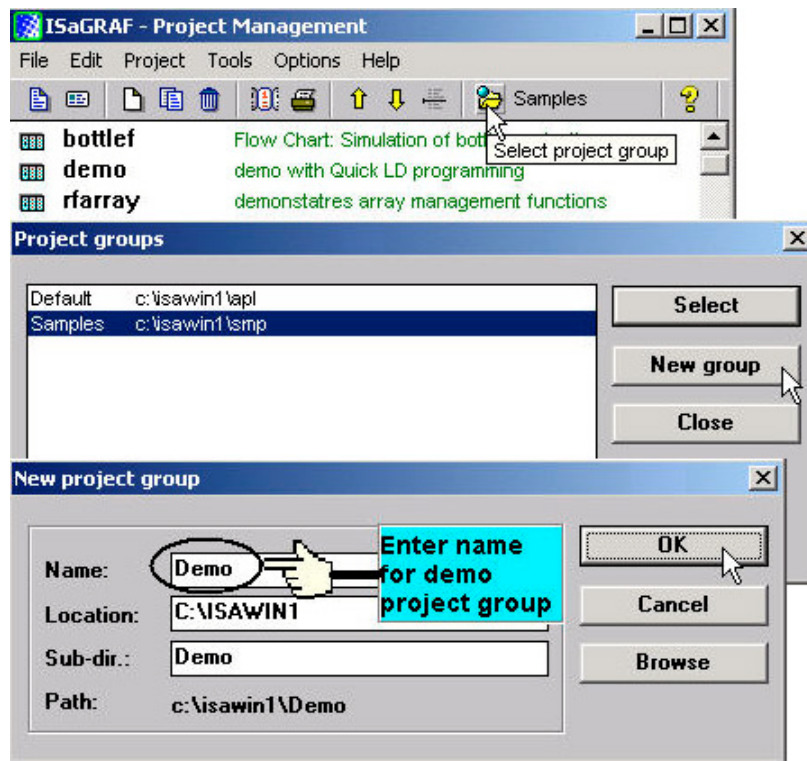
9.2: Installing The ISaGRAF Programming Examples

The ISaGRAF programming examples are installed on the same CD-ROM that you receive with the Wincon controller system. You will find the programming example files in the **\Napdos\ISaGRAF\Wincon\Demo** sub-directory on the CD-ROM.

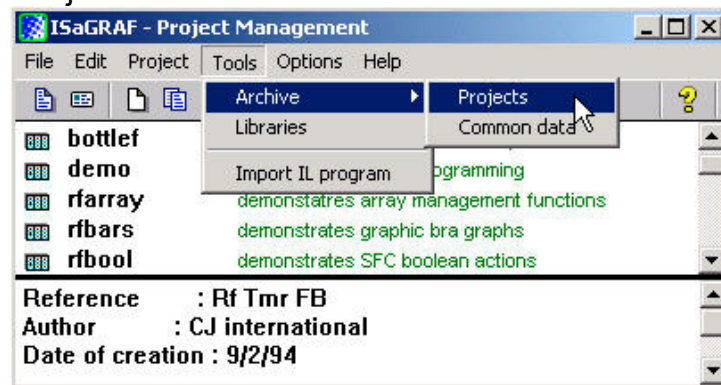
Example lists:

Project Name	Description	I/O Boards Or Complex Equipment Used
Wdemo_01	R/W float value from file	
Wdemo_02	R/W long integer value from file	
Wdemo_02a	R/W long integer value in Text format from file	
Wdemo_03	To output something at a scheduled time interval: For ex. Moday, 09:00 ~ 18:00, Sunday, 10:00 ~	
wdemo_04	User defined Modbus protocol (No using "Mbus")	
wdemo_05	To do something at some sec later when an event happens	i-8055
wdemo_06	Using Message Array - MsgAry_r , MsgAry_w	
wdemo_07	Convert float value to string, using real_str & rea_str2	
wdemo_08	PID control, refer to CD:\napdos\isgraf\wincon\english_manu\ "PID_A L...htm"	
wdemo_09	Store & backup boolean & long integer value To/From files	
wdemo_10	Store & backup boolean & long integer value To/From EEPROM	
wdemo_11	Connect to other Modbus RTU devices via COM2:RS232 or COM3:RS485	
wdemo_12a	Ebus_m	
wdemo_12b	Ebus_s	
wdemo_13	Send / receive Short Message from GSM Modem – "M1206"	
wdemo_14	Read / Write COM2 or COM3 or COM5 ...	
wdemo_15a	Bus7000 Redundant Master	
wdemo_15b	Bus7000 Redundant Slave	
wdemo_26	To move some pulse at x-axis of i-8091 of slot 1 in W-8337/8737	i-8091
wdemo_27	Motion x, slot 1: i-8091, slot 2: i-8090, Napdos\ISaGRAF\8000\Driver\motion.pdf	i-8091 i-8090
wdemo_28	Motion x-y, slot1: i-8091, slot2: i-8090, Napdos\ISaGRAF\8000\Driver\motion.pdf	i-8091 i-8090
wdemo_29	Moving to he Abs. position when CMD is given, slot 1 : i-8091, slot 2: i-8090	i-8091 i-8090

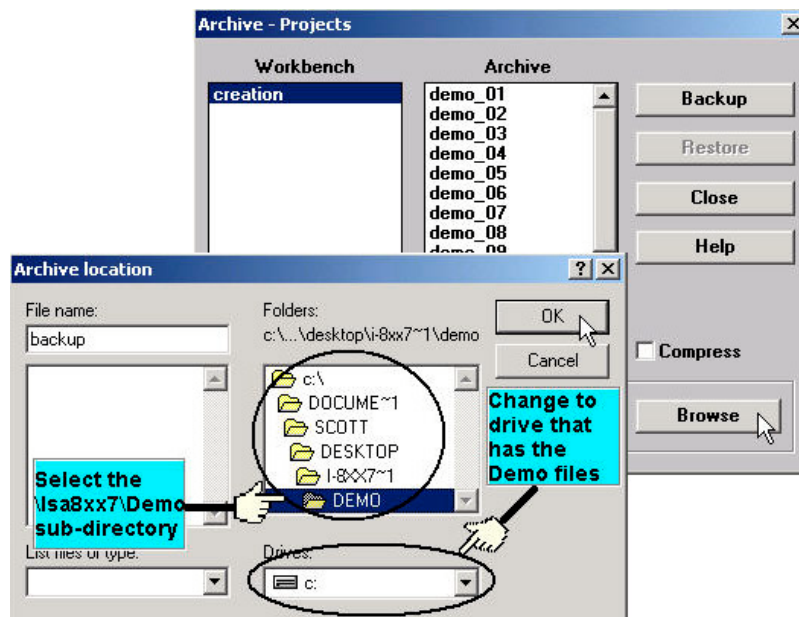
When you install the ISaGRAF programming example for the Wincon controller it is recommended that you create an "ISaGRAF Project Group" to install the demo program files into.



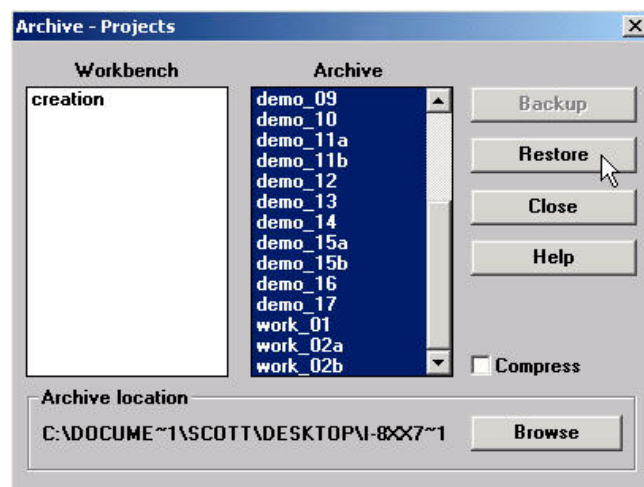
To install the demo programs into the project you have created open the "ISaGRAF Project Management" window to select "Tools" from the menu bar, then select the "Archive" option and then click on "Projects".



When you click on the "Projects" selection the "Archive Projects" window will open. Click on the "Browse" button to select the drive and the sub-directory where the demo files are located (\Napedos\ISaGRAF\Wincon\Demo\ on the CD-ROM) .



To install all of the Demo files, click on the "wdemo_01" file, then press and hold down the "Shift" key, continue to hold down the "Shift" key and use your mouse to scroll down to last file in the "Archive" window. Click on the last file name from the demo file location and that will select the entire group of demo files. Lastly, click on the "Restore" button in the "Archive Projects" window and all of the demo files will be installed into the sub-directory you have created.

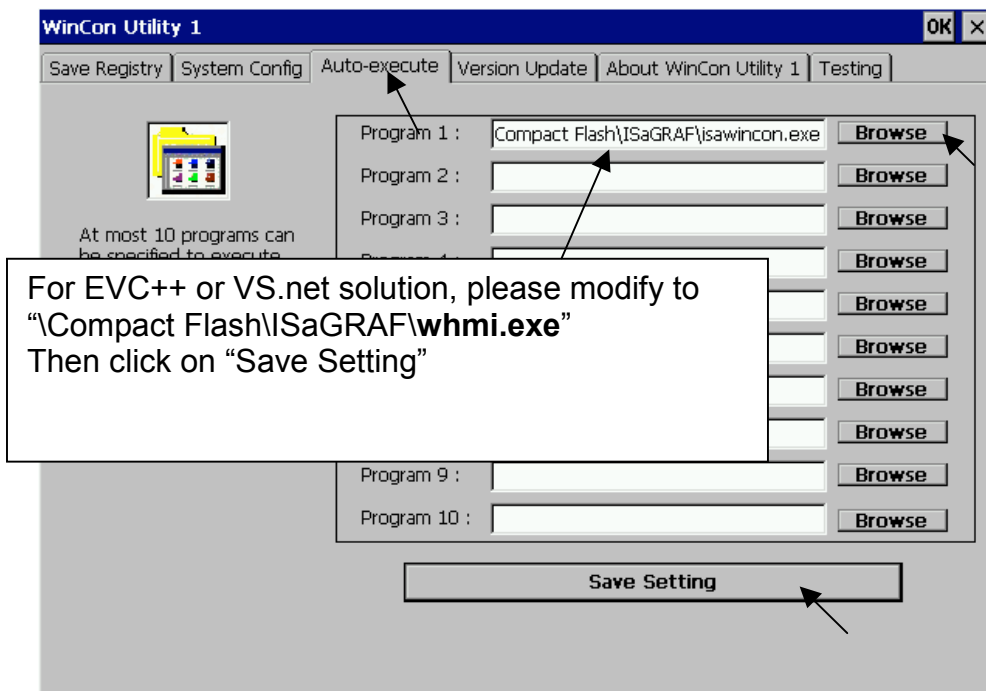


Chapter 10: Programming W-8x37 By Non-ISaGRAF

10.1: Set Whmi.exe To Be First Auto-execute Program

The Wincon-8037/8337/8737 supports ISaGRAF programming method & provides Web HMI solution by default by auto-running the “isawincon.exe” when controller is booting up.

If user prefer to program Wincon-8037/8337/8737 by using Microsoft EVC++ 4.0 or VS.net 2003 (VB.net & C#.net) and access to the Web HMI, it is also possible. In such a case, user don't need to purchase any ISaGRAF Software (ISaGRAF-256-C or ISaGRAF-256-E). Please run “Start” – “Programs” – “Wincon Utility” and then modify “auto-execute” to run “**whmi.exe**”.



Then save Registry. This step will re-boot your Wincon controller.



If user would like to program W-8x37 by using both ISaGRAF & (EVC++ or VS.net), it is also possible. Please refer to Chapter 6 or Chapter 7.

10.2: Setting Up An EVC++ Web HMI Demo

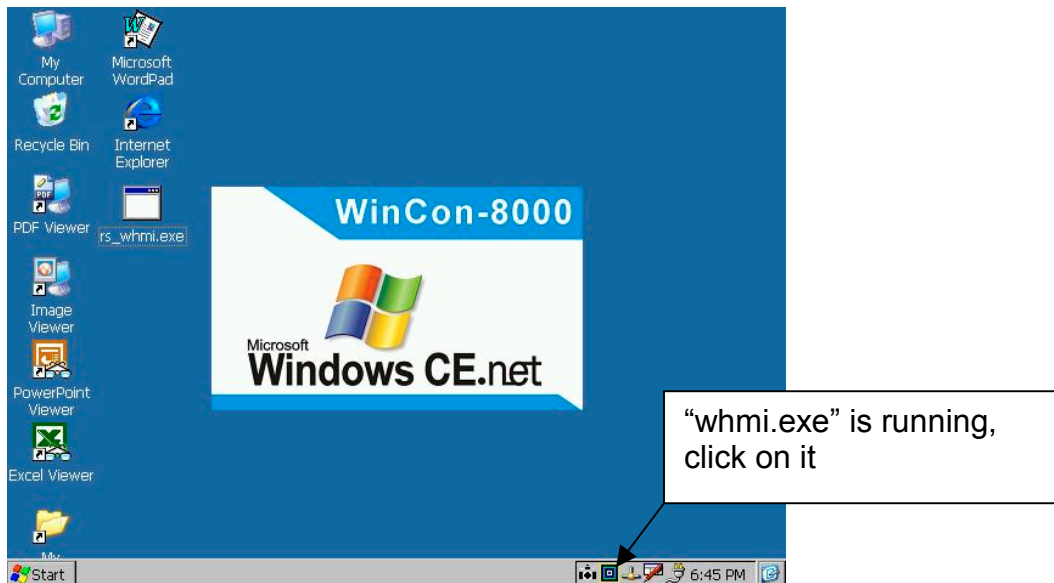
10.2.1: Step 1 - Setup The Hardware

- A. Please have a Wincon-8337 or Wincon-8737 & plug one I-8077 (or I-8055 or I-8054) board in its slot 1.
- B. Prepare one VGA monitor, one PS2 mouse, one Keyboard and one ethernet cable and then connect them to the Wincon.
- C. Power the Wincon up.

10.2.2: Step 2 - Make Sure Whmi.exe Installed

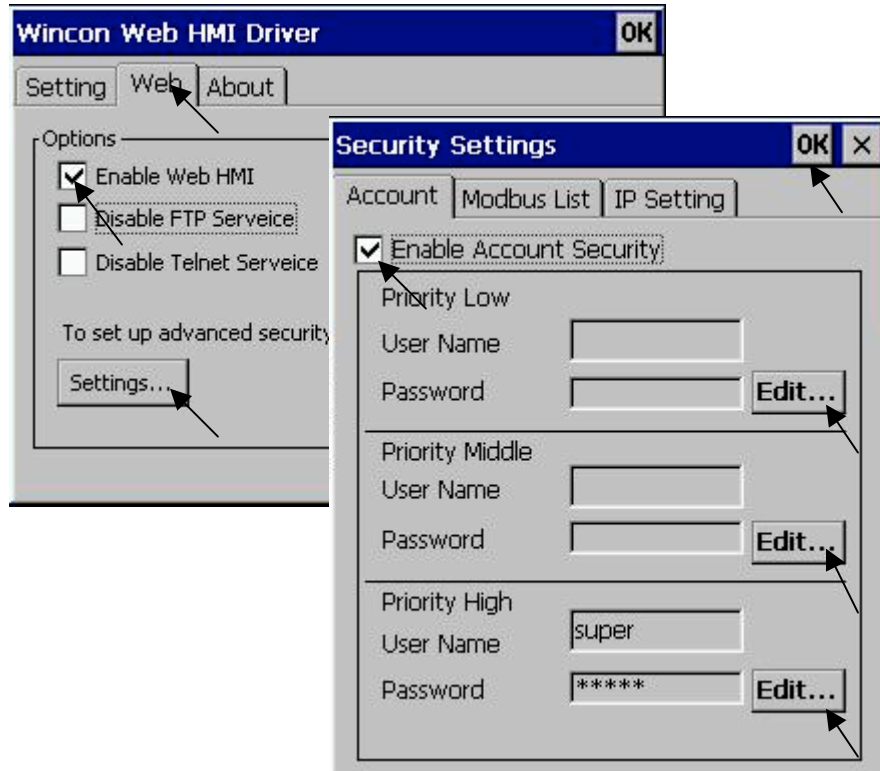
- A. Please set a fixed IP address to the Wincon-8037/8337/8737. (No more DHCP)
- B. Please refer to section 10.1 to set “whmi.exe” to be first auto-execute program & then reboot your Wincon.

10.2.3: Step 3 - Setting The Web Options



Check on “Enable Web HMI”, and then click on “Setting”, Please check on “Enable Account Security” ,then click on “Edit” to set (username , password). **Then remember to click on “OK”**

Note: If “Enable Account Security” is not check, any user can easily get access to your Wincon through the Internet Explorer.



10.2.4: Step 4 - Download And Run EVC++ Demo Program

Please download EVC++ Demo program form the

W-8x37 CD-ROM:

\napdos\isagraf\wincon\non_isagraf_demo\EVC++\whmi_c01\c\armv4rel\ "whmi_c01.exe"

to

Wincon: \Compact Flash\ISaGRAF\ and then run it.

10.2.5: Step 5 - Download Web Pages To The Wincon

A. Please copy all files in the CD-ROM:

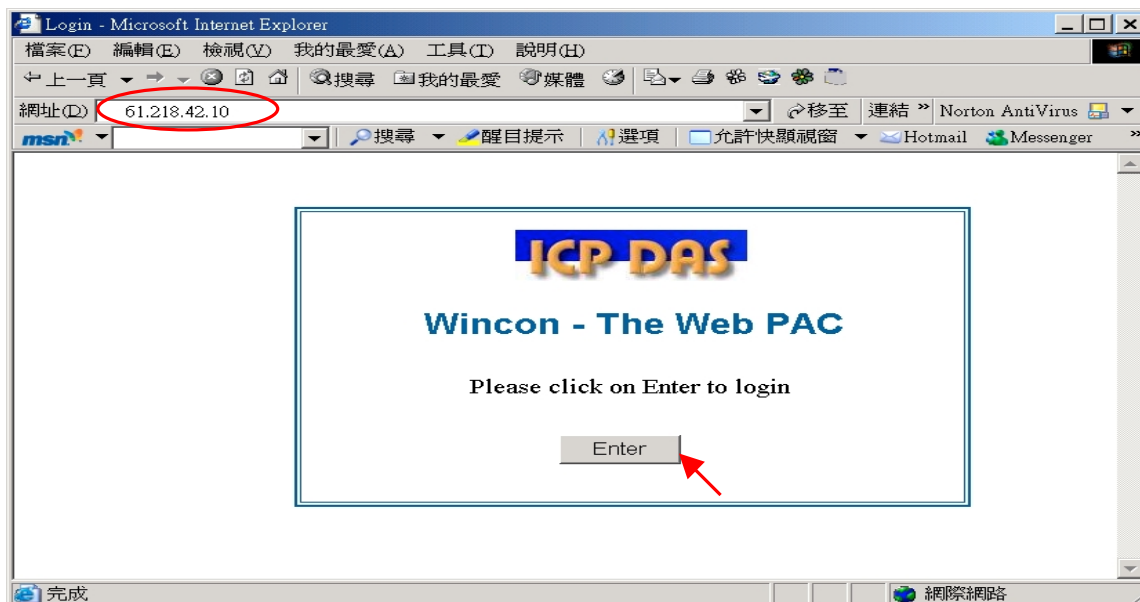
\napdos\isagraf\wincon\non_isagraf_demo\EVC++\whmi_c01\whmi\ *.* to
Wincon: \Compact Flash\Temp\HTTP\WebHMI\

B. Since the Web Pages are modified or new copied, please run “rs_whmi.exe” to reset the Web server. **The “rs_whmi.exe” must be run every time when user has modified any file in the W-8x37’s Compact Flash\Temp\HTTP\WebHMI**



10.2.6: Step 6 - Show Time

Please run Internet Explorer (Rev. 6.0 or higher), key in the IP address of your W-8037/8337/8737. For example: 61.218.42.10 or http://61.218.42.10



10.3: EVC++ Function To Access To The Web HMI

User can write his EVC++ 4.0 application to access to the Web HMI. The include file and library are "WinConAgent.h" and "Quicker.lib". (CD-ROM:\napdos\isagraf\wincon\EVC++_LIB\). The DLL at run time is the "Quicker.dll" which is in Wincon - \Compact Flash\ISaGRAF\ (Please copy and run your EVC++ application at Wincon - \Compact Flash\ISaGRAF\). The below functions are used for access to the Web HMI data base. You may refer to CD-ROM:\napdos\isagraf\wincon\non_isagraf_demo\EVC++\ for example programs.

Note: Please check "whmi.exe" listed in section 10.1 has been installed for non_ISaGRAF solution.

Set boolean value to Web HMI:

unsigned char UserSetCoil(unsigned short iUserAddress, unsigned char iStatus);

iUserAddress:

1 to 1024 (Web HMI's boolean data base)

2001 to 3024 (indicates if Internet Explorer post writing command to W-8x37)

iStatus: 0: set boolean to False, 1: set boolean to True

for ex,

```
long temp_val;
```

```
// please set starting & ending Addr No. of each type at the beginning of your C program
```

```
temp_val = 100 ;
```

```
UserSetReg(2001 , &temp_val, 0) ; // set boolean's starting addr. To 100
```

```
temp_val = 200 ;
```

```
UserSetReg(2002 , &temp_val, 0) ; // set boolean's ending addr. To 200
```

```
...
```

```
UserSetCoil(100 , 1) // set boolean at network addr 100 as True
```

Note : The boolean's Address No. 2001 to 3024 indicates if the IE on the PC gives writing command to the Web HMI's data base. For example, If IE gives writing command to set Address 1 & 2's boolean value to True (or False) and set Address 100's float value to 1.234 (or any float value) and set Address 1009's word value to -456 (or any word value), then the boolean value in Address 2001 & 2002 & 2100 & 3009 will set to 1 by the Web HMI. This means IE has given writing command to address 1 & 2 & 100 & 1009. User's EVC++ program can get the information (by using **UserGetCoil()** to get boolean value in address 2001 to 3024) to know if "IE on the PC want to set value to the controller". **Please be very careful to call UserSetCoil() to clear it** (set to 0 to the related Address in Address 2001 to 3024), **if your C program has got the information**. The Web HMI driver will set flag value to 1 next time when IE post writing command.

Get boolean value from Web HMI:

unsigned char UserGetCoil(unsigned short iUserAddress, unsigned char *iStatus);

iUserAddress: **1 to 1024** (Web HMI's boolean data base)

2001 to 3024 (indicates if Internet Explorer post writing command to W-8x37)

iStatus: 0: boolean is False, 1: boolean is True

for ex.

unsigned char bVal, flag_bVal;

...

UserGetCoil(2005 , &flag_bVal) // get information flag of addr 5

UserGetCoil(5 , &bVal) // get boolean value at network addr 5

UserSetCoil(2005 , 0) // clear the information flag of Address 5

Set word or float or long value to Web HMI:

unsigned char UserSetReg(unsigned short iUserAddress, long *iStatus, unsigned char iDType);

iUserAddress:

1 to 1024 (Web HMI's analog data base)

2001 : starting Address No. of **boolean** data base to send to the IE on PC.

2002 : End Address No. of **boolean** data base.

2003 : starting Address No. of **word** data base to send to the IE on PC.

2004 : End Address No. of **word** data base.

2005 : starting Address No. of **float** data base to send to the IE on PC.

2006 : End Address No. of **float** data base.

2007 : starting Address No. of **long** data base to send to the IE on PC.

2008 : End Address No. of **long** data base.

2009 : starting Address No. of **string** data base to send to the IE on PC.

2010 : End Address No. of **string** data base.

iStatus: A pointer to a long type, which stores the data to set

iDType 0: type is word

1: data type is float

2: data type is long

Note:

1. The Address No. for each type must not conflict with each other. For ex, below assignment is OK. Boolean:(1 to 100), Word:(101 to 120), Float:(121 to 130), long:(131 to 132), String:(201 to 202). However, Boolean:(1 to 100), Word:(101 to 120), Float:(121 to 130), long:(125 to 132), String:(201 to 202) is not Ok, because Float & long conflict at No. of 125 to 130.

2. The more No. assigned will make the data size bigger to deliver to IE. Please do not assign too many No. especially for float, long & string data.

for ex.

```
float  float_val;
long   word_val, long_val[2];
long   temp_val;
long   *long_pt;

// please set starting & ending Addr No. of each type at the beginning of your C program
// set word starting addr. To 1
temp_val = 1;
UserSetReg(2003 , &temp_val, 0) ;
// set word ending addr. To 10
temp_val = 10 ;
UserSetReg(2004 , &temp_val, 0) ;
// set float's starting addr. To 11
temp_val = 11 ;
UserSetReg(2005 , &temp_val, 0) ;
// set float's ending addr. To 20
temp_val = 20 ;
UserSetReg(2006 , &temp_val, 0) ;
// set long's starting addr. To 21
temp_val = 21;
UserSetReg(2007 , &temp_val, 0) ;
// set long's ending addr. To 30
temp_val = 30 ;
UserSetReg(2008 , &temp_val, 0) ;
...

// set word_val (-32768 to +32767) to network address 1
word_val = -20000 ;
long_pt = (long *)&word_val ;
UserSetReg(1 , long_pt, 0);

// set float_val to network address 11
float_val = 1.2345 ;
long_pt = (long *)&float_val;
UserSetReg(11 , long_pt, 1);

// set long_val to network address 21
long_val[0] = 12345678 ;
long_pt = (long *)&long_val[0] ;
UserSetReg(21 , long_pt, 2);

// set long_val to network address 23
long_val[1] = -4567 ;
long_pt = (long *)&long_val[1] ;
UserSetReg(23 , long_pt, 2);
```


Get word or float or long value from Web HMI:

`unsigned char UserGetReg(unsigned short iUserAddress,long *iStatus,unsigned char iDType);`

iUserAddress:1 to 1024 (Web HMI's analog data base)
iStatus: A pointer to a long type, which will store the data returned
iDType 0: type is word
1: data type is float
2: data type is long

for ex.

```
float float_val;  
long word_val, long_val[2];  
long ret_val;  
unsigned char flag_float_val, flag_word_val, flag_long_val[2] ;  
  
...  
  
// get word_val (-32768 to +32767) of network address 1  
UserGetCoil(2001 , &flag_word_val) // get information flag of addr. 1  
UserSetCoil(2001 , 0) // clear the information flag of Address 1  
UserGetReg(1, &ret_val, 0);  
if ( ret_val>=0 && ret_val<=32767 ) word_val = ret_val;  
else word_val = ret_val | 0xFFFF0000;  
  
// get float value of network address 11  
UserGetCoil(2011 , &flag_float_val) // get information flag of addr. 11  
UserSetCoil(2011 , 0) // clear the information flag of Address 11  
UserGetReg(11, &ret_val, 1) ;  
float_val = *(float *) (&ret_val) ;  
  
// get long value of network address 21  
UserGetCoil(2021 , &flag_long_val[0]) // get information flag of addr. 21  
UserSetCoil(2021 , 0) // clear the information flag of Address 21  
UserGetReg(21, &ret_val, 2) ;  
long_val[0] = ret_val ;  
  
// get long value of network address 23  
UserGetCoil(2023 , &flag_long_val[1]) // get information flag of addr. 23  
UserSetCoil(2023 , 0) // clear the information flag of Address 23  
UserGetReg(23, &ret_val, 2) ;  
long_val[1] = ret_val ;
```

Set string value to Web HMI:

```
unsigned char UserSetReg_Str(unsigned short iUserAddress, char *iStatus);
```

iUserAddress: 1 to 1024 (Web HMI's string data base)

iStatus: A pointer to a char[], which stores the string to set. Max string len is 255 bytes.

Get string value from Web HMI:

```
unsigned char UserGetReg_Str(unsigned short iUserAddress, char *iStatus);
```

iUserAddress: 1 to 1024 (Web HMI's string data base)

iStatus: A pointer to a char[], which will store the string returned. Max string len is 255 bytes.

For example,

```
char str0[256] , str1[256];  
long temp_val;  
unsigned char flag_str0 ;
```

```
// please set starting & ending Addr No. of each type at the beginning of your C program
```

```
// set string's starting addr. to 201
```

```
temp_val = 201 ;
```

```
UserSetReg(2009 , &temp_val, 0) ;
```

```
// set string's ending addr. to 202
```

```
temp_val = 202 ;
```

```
UserSetReg(2010 , &temp_val, 0) ;
```

```
...
```

```
UserGetCoil(2201 , &flag_str0) // get information flag of addr. 201
```

```
UserSetCoil(2201 , 0) // clear the information flag of Address 201
```

```
// get string of network address 201
```

```
UserGetReg_Str(201, str0);
```

```
// Set string to network address 202
```

```
UserSetReg_Str(202, str1);
```

Note:

1. The Address No. for each type must not conflict with each other. For ex, below assignment is OK. Boolean:(1 to 100), Word:(101 to 120), Float:(121 to 130), long:(131 to 132), String:(201 to 202). However, Boolean:(1 to 100), Word:(101 to 120), Float:(121 to 130), long:(125 to 132), String:(201 to 202) is not Ok, because Float & long conflict at No. of 125 to 130.

2. The more No. assigned will make the data size bigger to deliver to IE. Please do not assign too many No. especially for float, long & string data.

10.4: VB.NET Function To Access To The Web HMI

You may refer to

1. CD-ROM:\napdos\isagraf\wincon\non_isagraf_demo\VB.NET\ for example programs.
2. Section 10.2 for similar steps to set up VB.NET demo

vb demo program:

\napdos\isagraf\wincon\non_isagraf_demo\VB.NET\whmi_vb01\vb01\bin\release\ "vb01.exe"

Web pages:

\napdos\isagraf\wincon\non_isagraf_demo\VB.NET\whmi_vb01\whmi\ *.*

Note: Please check "whmi.exe" section 10.1 has been installed for non_ISaGRAF solution.

User can write his VB.NET application to access to the Web HMI. The Address rule is the same as the former section. The Address No. for each type must not conflict with each other.

Web HMI boolean data base:	Address No. 1 to 1024
Web HMI word data base:	Address No. 1 to 1024
Web HMI float data base:	Address No. 1 to 1024
Web HMI long data base:	Address No. 1 to 1024
Web HMI String data base:	Address No. 1 to 1024

Information Flag: boolean Address No. **2001** to **3024**

Word Address No. of

2001 : starting Address No. of **boolean** data base to send to the IE on PC.
2002 : End Address No. of **boolean** data base.
2003 : starting Address No. of **word** data base to send to the IE on PC.
2004 : End Address No. of **word** data base.
2005 : starting Address No. of **float** data base to send to the IE on PC.
2006 : End Address No. of **float** data base.
2007 : starting Address No. of **long** data base to send to the IE on PC.
2008 : End Address No. of **long** data base.

Please refer to section 6.6 for the VB.NET function to access to Web HMI.

"UserSetCoil" , "UserGetCoil" to access to Web HMI's boolean data.

"UserSetReg_short" & "UserGetReg_short" to access to Web HMI's word data.

"UserSetReg_float" & "UserGetReg_float" to access to Web HMI's float data.

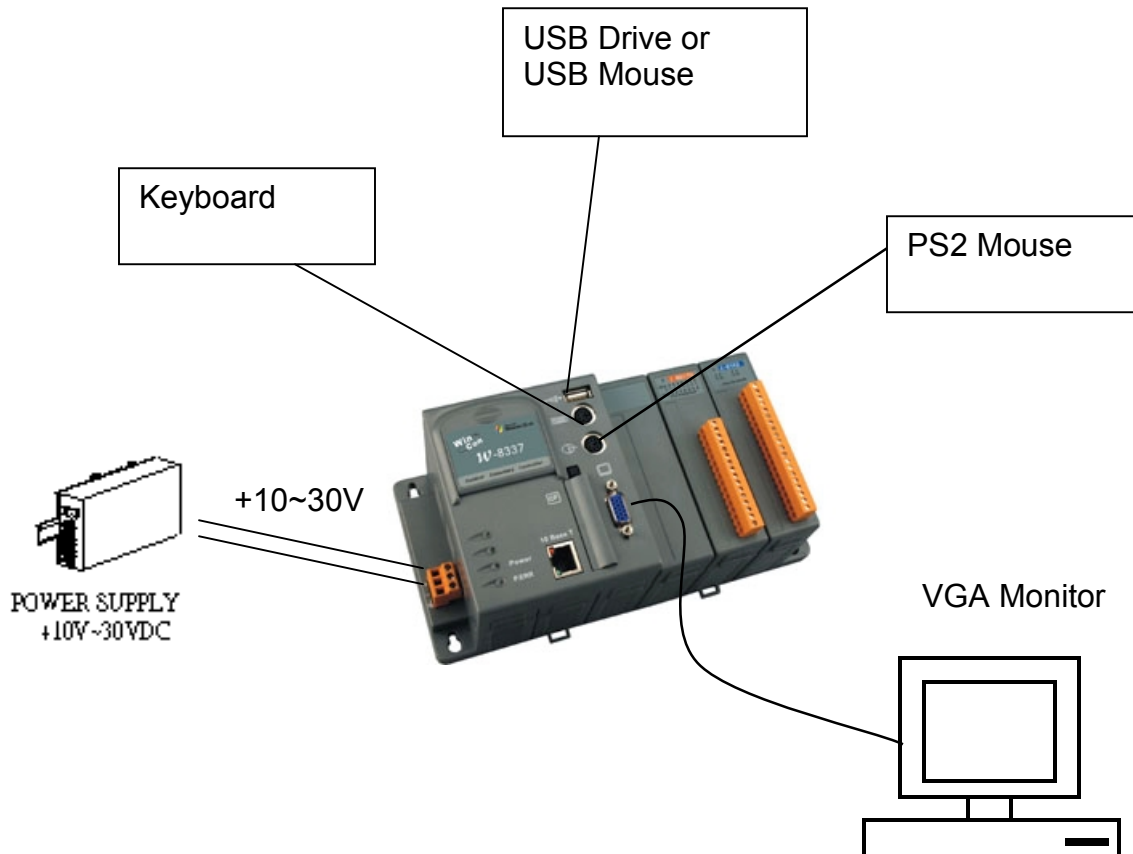
"UserSetReg_long" & "UserGetReg_long" to access to Web HMI's long integer data.

Note: VB.net & C#.NET can not access to string data of the Web HMI.

Appendix A: Hardware System & Setting

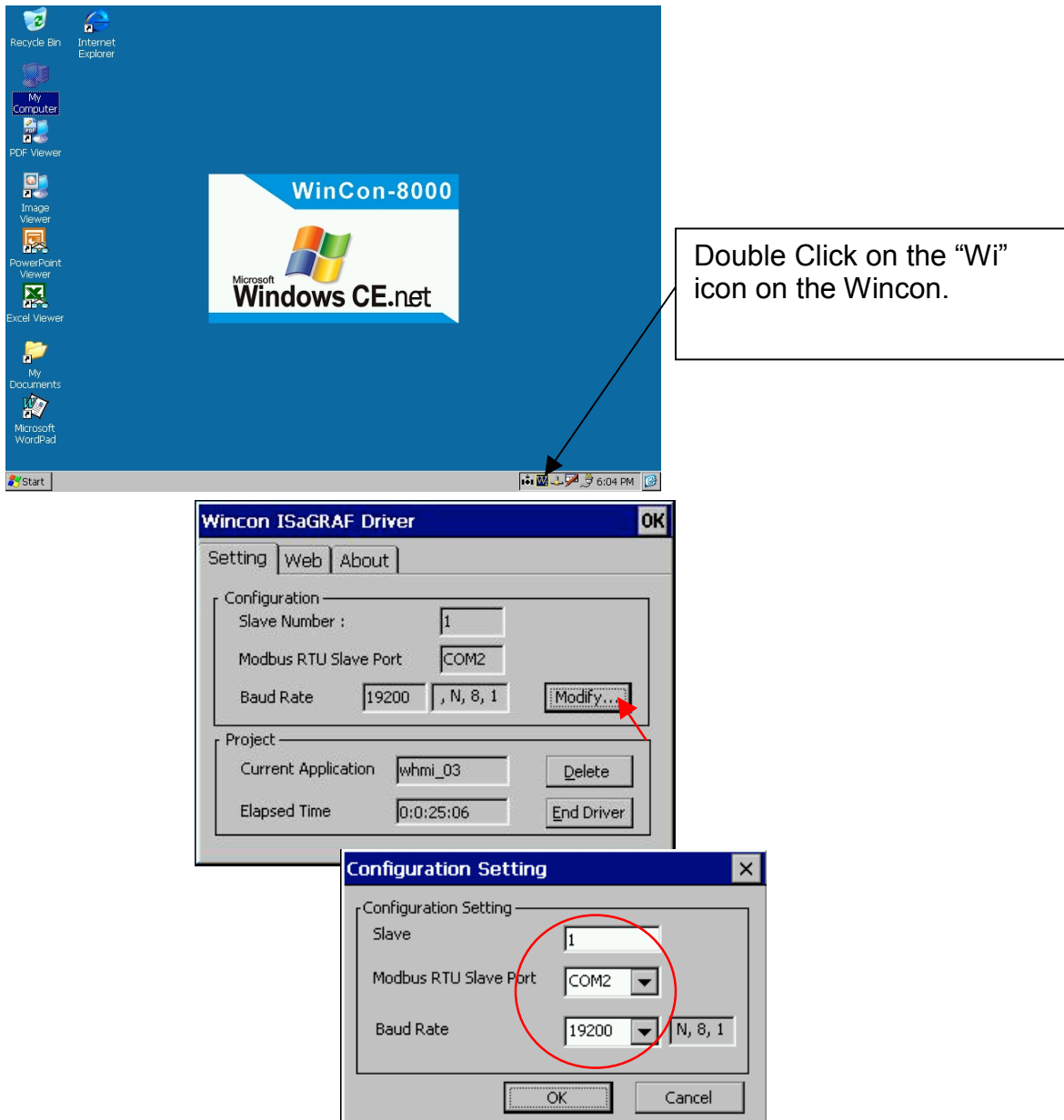
A.1: Applying Correct Power Supply

Please apply a power supply between +10V to +30V (> 25W or higher is better)



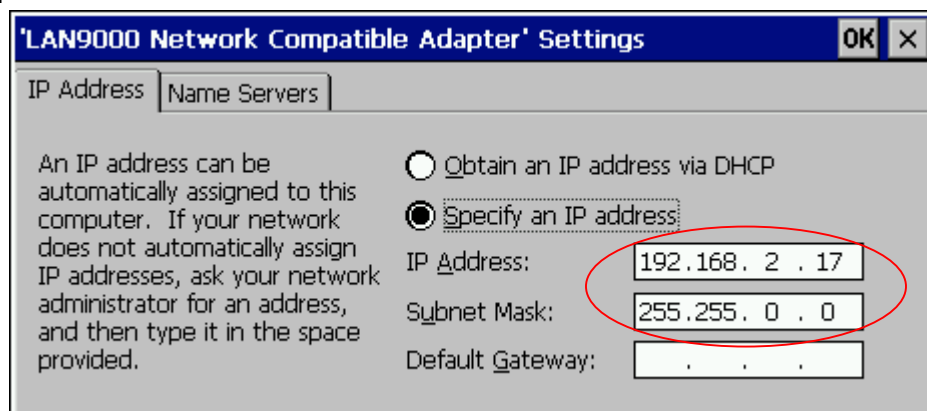
A.2: Modify The NET-ID & Modbus RTU Port Setting

User may set wincon-8037/8337/8737's Net-ID (Slave Number) to a No. from 1 to 255.
The default Modbus RTU port is "COM2". User may set it to others depends on its application.

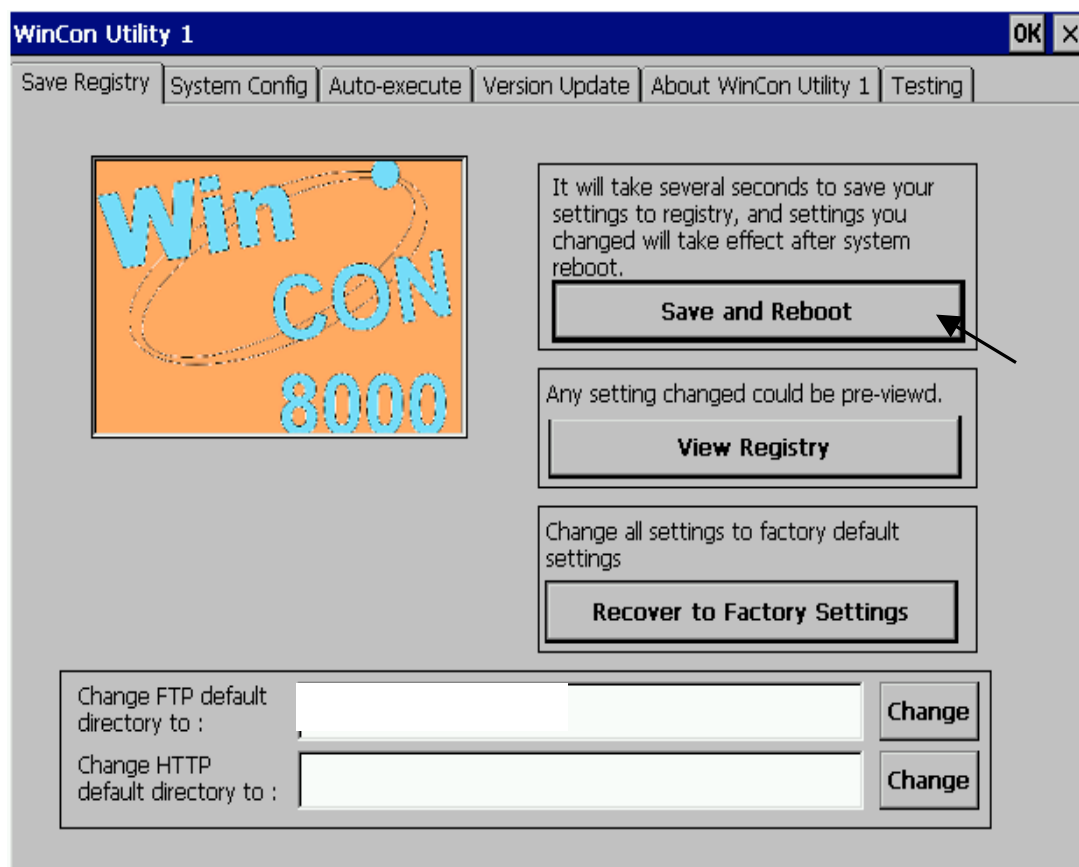


A.3: Setting The IP Address For The W-8x37

Please run “Start” – “Setting” – “Control Panel” on the Wincon, then double click on “Network and Dial-up Connections”. Then click on “LAN90001”. Set your Wincon’s IP address & its Subnet Mask.



Please run “Start” – “Programs” – “Wincon Utility”, click on “Save and Reboot”



A.4: Connecting Your PC To The W-8x37 Ethernet Port

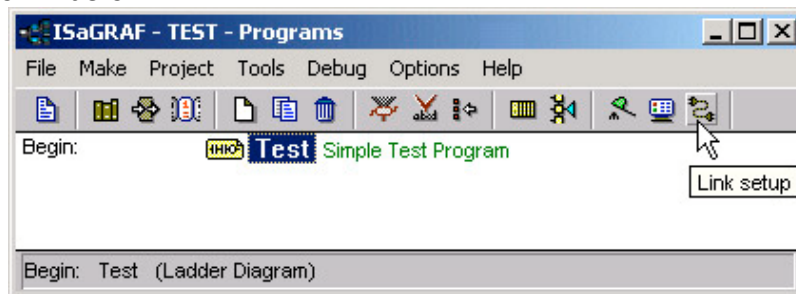
Before you can download an ISaGRAF application to the W-8x37 controller system using the Ethernet port, you must first setup the Ethernet port to properly communicate with the host PC.

On the W-8x37, Set IP, Mask and Gateway address:

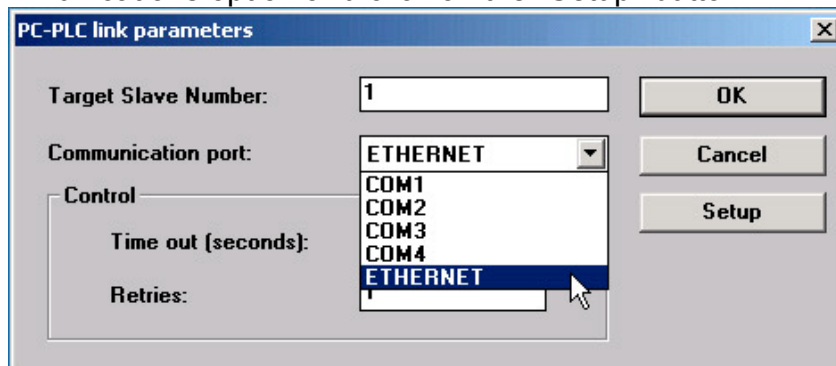
Refer to former section – “A.3: Setting The IP Address For The W-8x37”

On your PC:

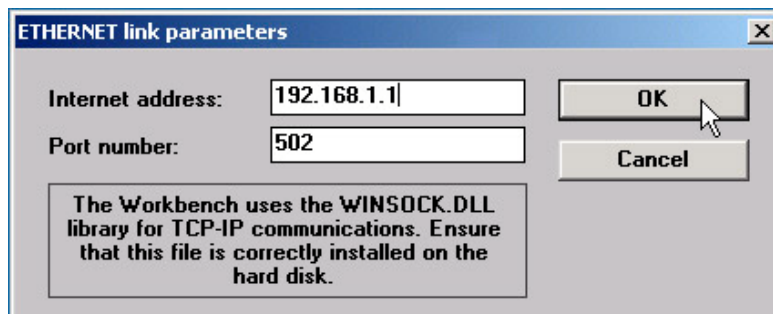
First open an ISaGRAF project and select a program you wish to communicate between your PC and the W-8x37 controller system. Next, select the "Link Setup" button on the project screen as shown below.



A "PC-PLC Link Parameters" dialog box will appear as shown below. From here select the "Ethernet" communications option and click on the "Setup" button.



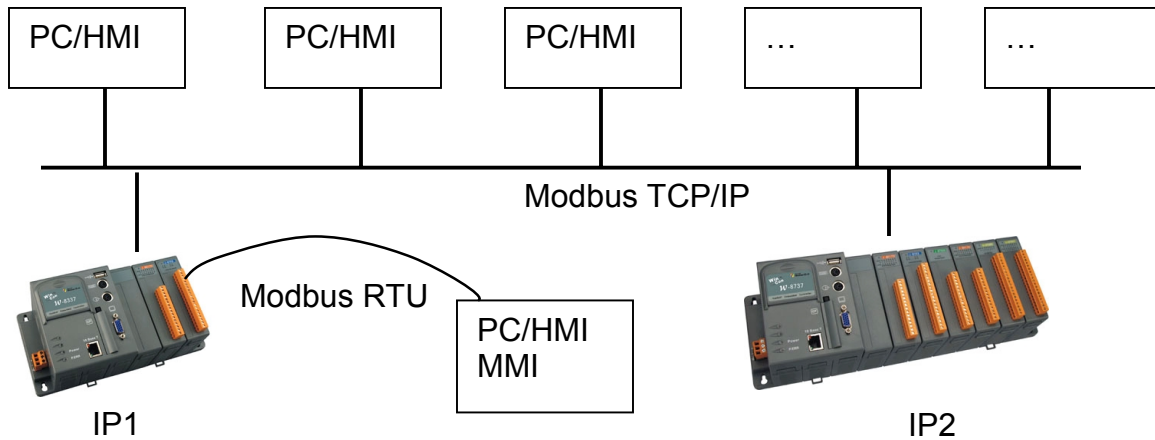
Once you have clicked on the "Setup" button, an "Ethernet Link Parameters" dialog box will appear. Set the "Port Number" to "502" and enter in the **Internet address (IP) of the W-8x37 controller**.



Once you have entered the appropriate information, click on the "OK" button, and now you have configured your PC to communicate with the W-8x37 through the Ethernet port.

A.5: Multi-Clients Connection to The W-8x37 Ethernet Port

Each W-8037/8337/8737 has an IP address and with a fixed Ethernet port No. **502**. Up to 8 PCs can link to one W-8037/8337/8737 throughout Ethernet (Modbus TCP/IP protocol). Another PC or MMI can link to COM2: RS232 port or COM3: RS485 (Modbus RTU protocol).

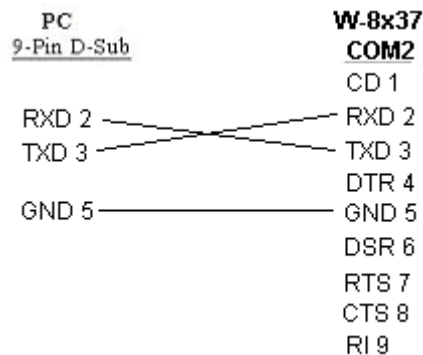


A.6: Connecting Your PC To The W-8x37 COM2 or COM3

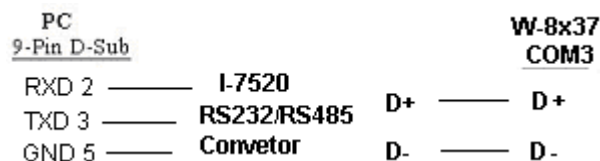
The default Modbus RTU port is COM2:RS232. User may change it to “COM2:RS232” or “COM3:RS485” or “None”. (please refer to “A.2: Modify The NET-ID & Modbus RTU port setting”)

Default communication parameter is “19200,8,N,1”

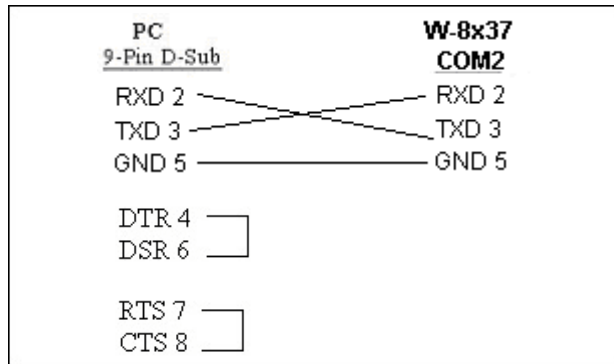
Pin assignment of the Wincon’s COM2: RS232 is the same as PC’s COM1. So the cable should be 2, 3 cross.



If connecting your PC to Wincon’s COM3:RS485, you need a I-7520 (RS232/485 converter) as below.



For the ISaGRAF Workbench RS-232 communications to operate properly, only the RXD, TXD, and the GND signals are used. If your PC is running a hardware device or software program that uses the CTS and DSR signals, you will need to wire the RTS-CTS and DTR-DSR signals together as shown below.

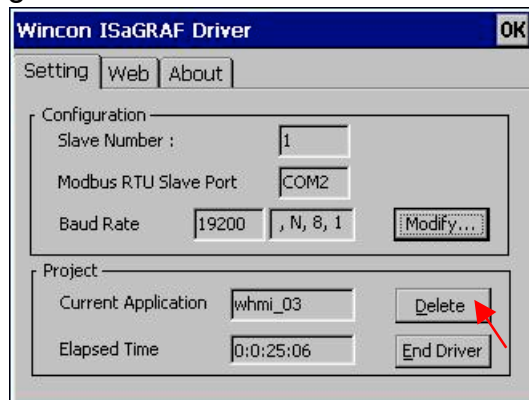


A.7: Deleting the ISaGRAF Project From The W-8x37

For some reasons, user may delete the ISaGRAF program in the Wincon-8x37 controller.



Click on "Setting" & then click on "Delete ISaGRAF Project".

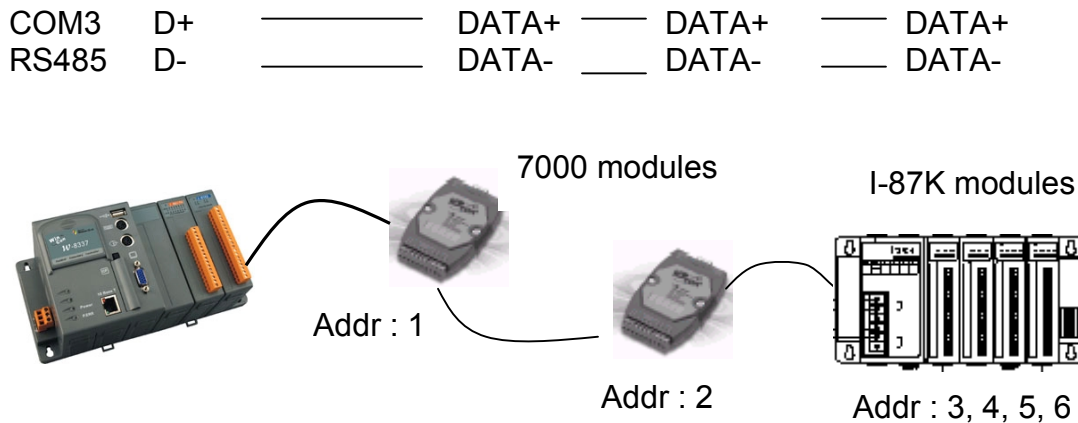


A.8: Linking I-7000 and I-87K Modules For Remote I/O

The W-8x37 controller system can use its COM3 port to link to ICP DAS's "I-7000" and "I-87K" series of remote I/O modules. This configuration can be very useful in applications that require distributed remote I/O throughout the system.

You can link up to **255** I-7000 or I-87K series remote modules to one W-8x37 controller system (It is better not to link up to 40 I-7000 or I-87K). You must remember to set each I-7000 and I-87K remote module must have a unique address, and be set to the same baud rate as the W-8x37 controller system.

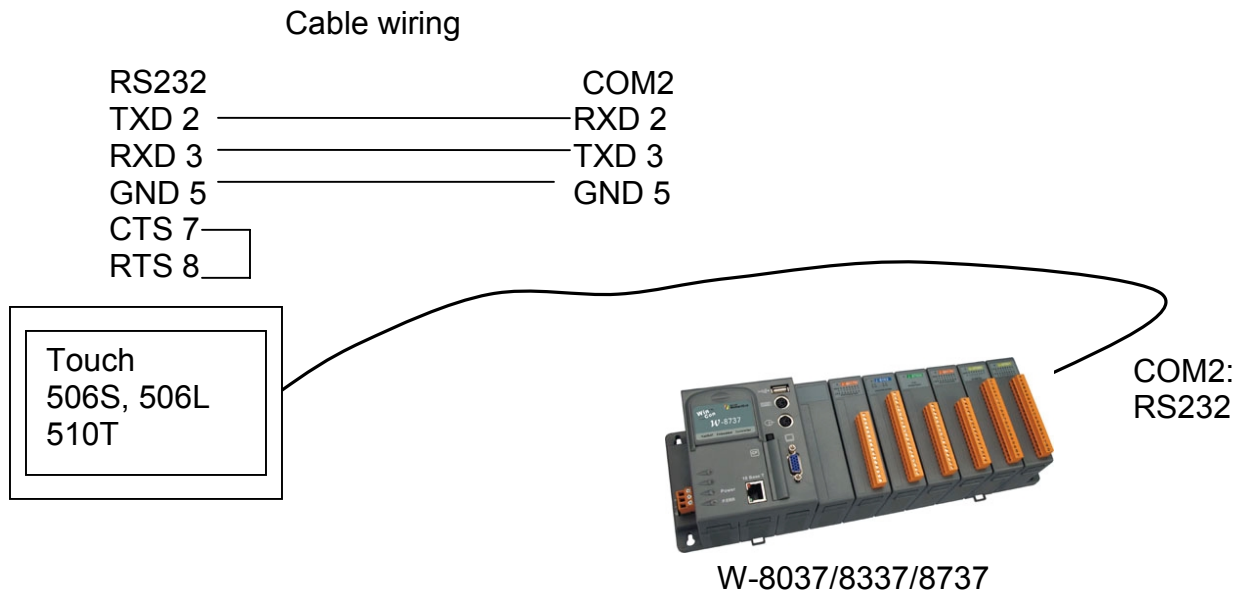
For more information regarding setting up and programming an I-7000 / I-87K remote module, please refer to Chapter 6 - "Linking To I-7000 and I-87K Modules" of the "User's Manual Of The ISaGRAF Embedded Controller" .



A.9: Linking To An HMI Interface Device

The COM2 (RS-232) and COM3 (RS-485) ports of the W-8x37 controller system can be used to interface with additional Human Machine Interface (HMI) devices such as touch screen displays. Please refer to section A.2 first for setting Modbus RTU port. ICP DAS provides a full line of touch screen displays, such as the "Touch" series screens. The models in the product line include the Touch 506S, Touch 506L, and Touch 510T HMI products.

If you are using any of the "Touch" series of MMI devices to connect to an W-8x37 controller, you can only interface the devices to the COM2 port on the W-8x37 controller.

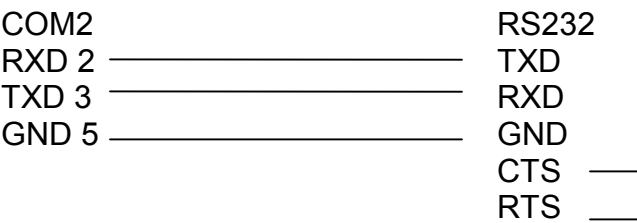
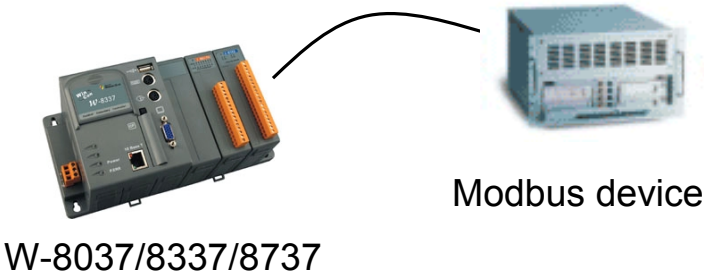


For more information regarding interfacing the Touch series of MMI devices to the W-8x37 controller system, please refer to Chapter 4- "Linking The I-8xx7 To HMI Devices" of the "User's Manual Of The ISaGRAF Embedded Controller" ..

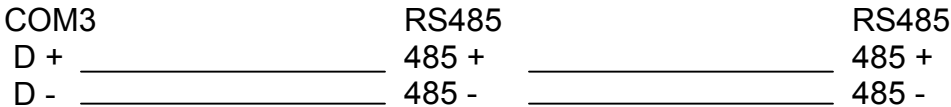
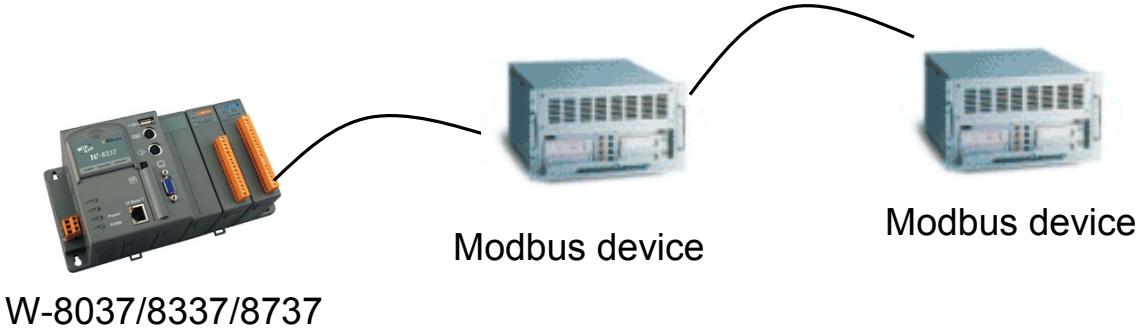
A.10: Linking To Other Modbus Devices

The COM2 (RS-232) or COM3 (RS-485) supports Modbus Master protocol. Please refer to Chapter 8 of the “User’s Manual Of The ISaGRAF Embedded Controllers” for more information.

COM2 (RS232) :



COM3 (RS485) :



Appendix B: Update Wincon's ISaGRAF Driver to Higher Version

Note:

If you have purchased W-8037 or W-8337 or W-8737 or W-8036 or W-8336 or W-8736, the Wincon ISaGRAF Driver is already installed when shipping out. You don't need to install it. However if you want to upgrade to higher version, you may upgrade it by yourself.

The Wincon ISaGRAF driver can be obtained at

WINCON8000 CD-ROM: \napdos\isagraf\wincon\driver\<version Number>\

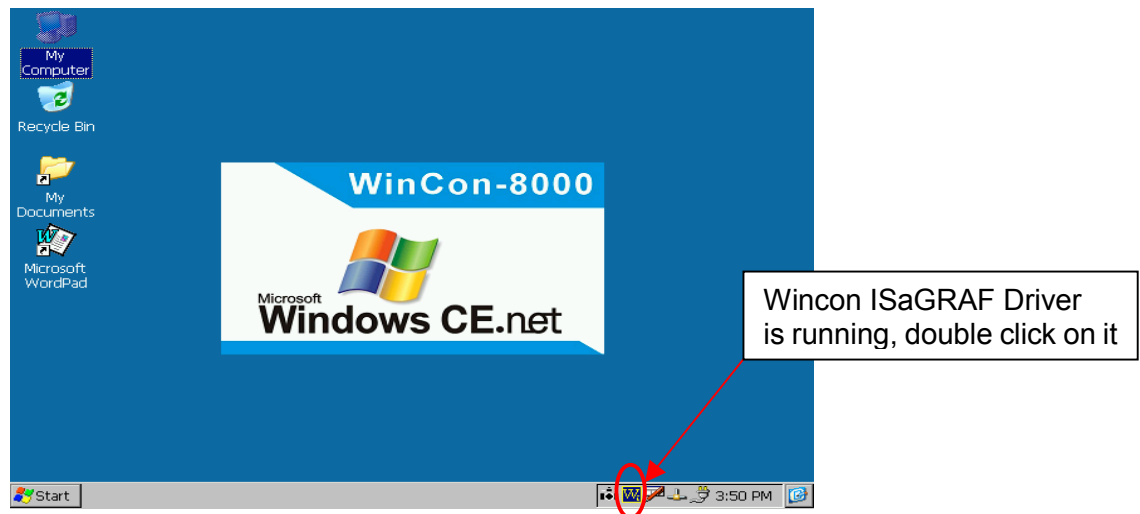
For example, version 3.12 is located at

\napdos\isagraf\wincon\driver\3.12\

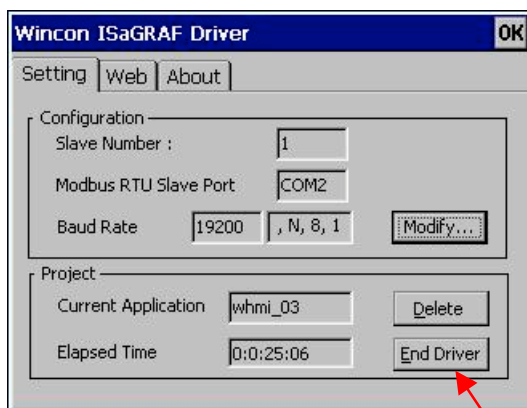
Or you may download it from

<http://www.icpdas.com/products/8000/isagraf.htm>

1. If your Wincon is W-8037/8337/8737 or W-8036/8336/8736, please stop "Wincon ISaGRAF Driver" first. However if it is W-8031/8331/8731 or W-8039/8339/8739, please goto step 2.

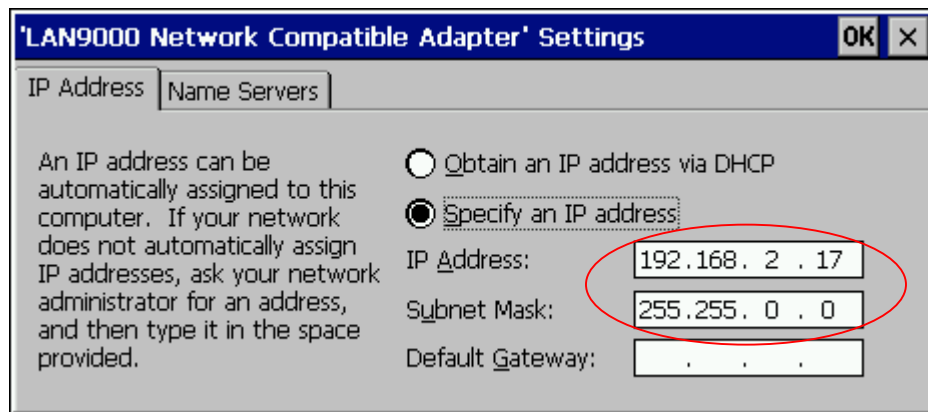


Click on "End Driver" to stop it.

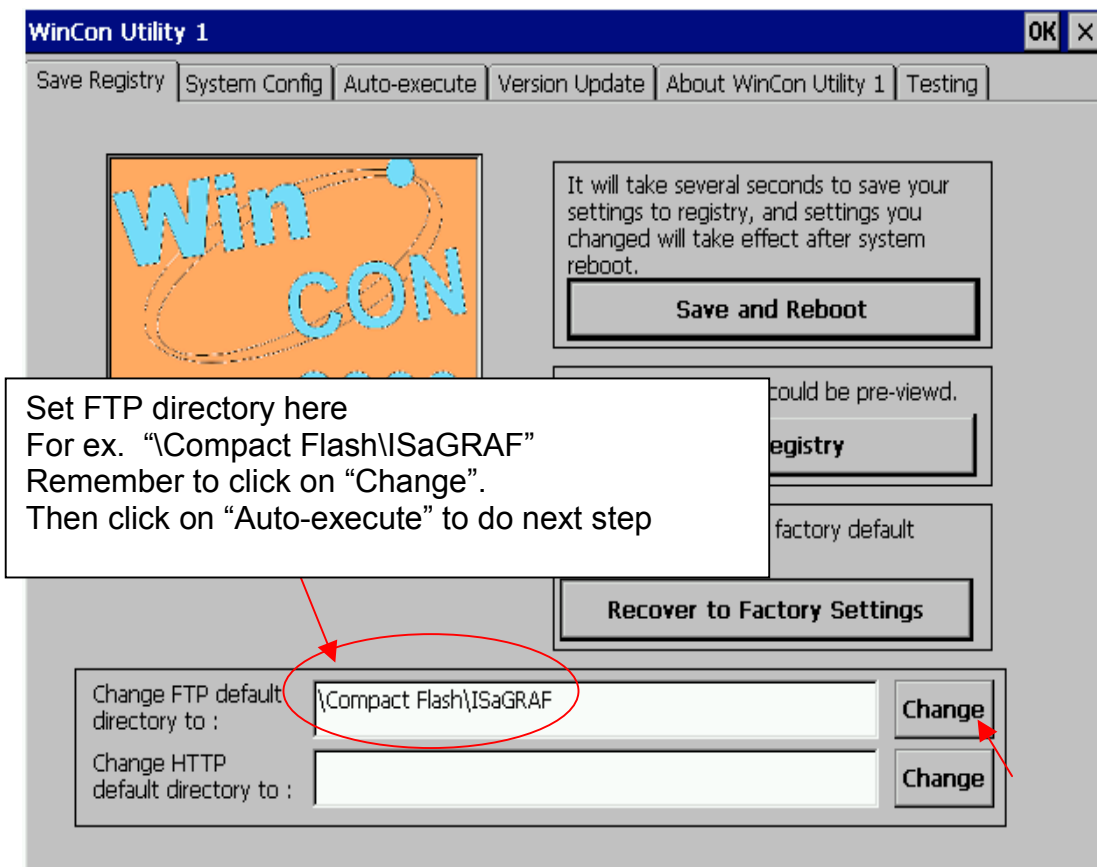


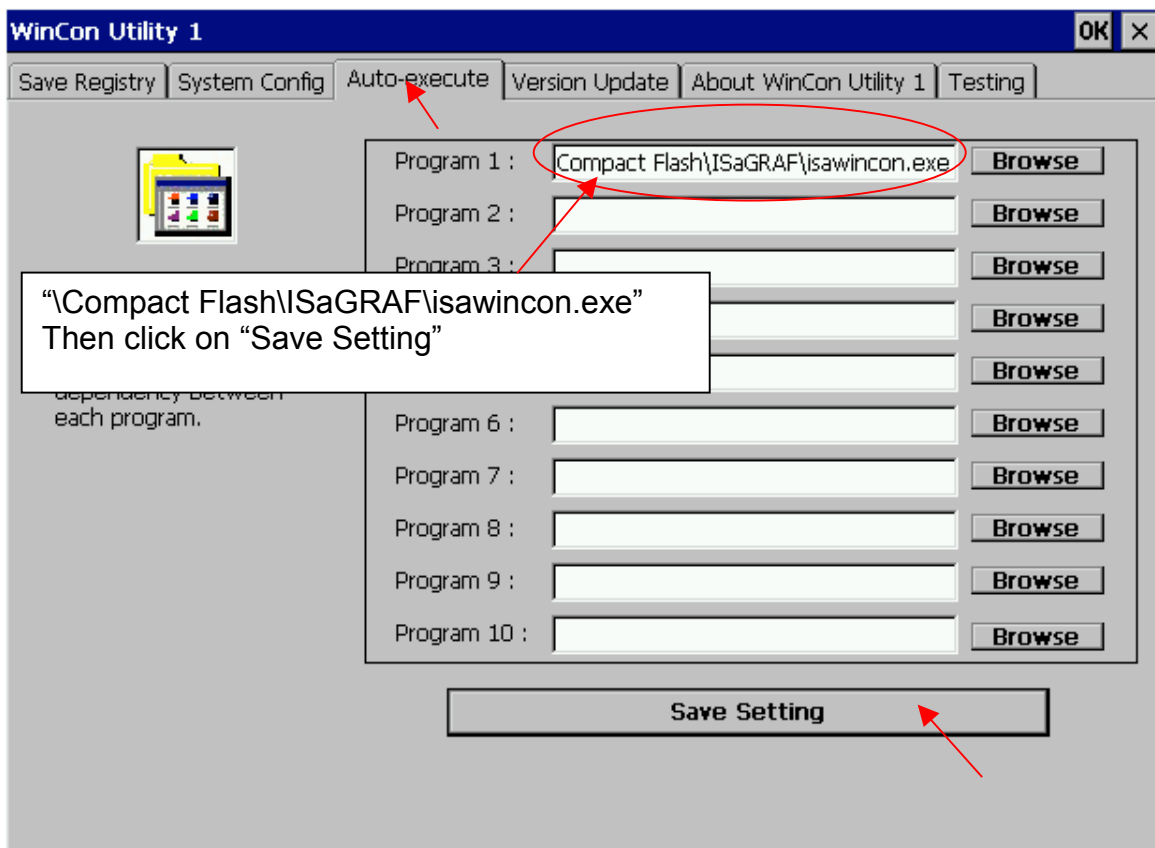
2. Set up Wincon's IP, Mask, FTP directory & Auto-execute

- A. Please create a folder "ISaGRAF" inside "\Compact Flash" folder in your Wincon controller. Then it will be \Compact Flash\ISaGRAF
- B. Please run "Start" – "Setting" – "Control Panel", then double click on "Network and Dial-up Connections". Then click on "LAN90001". Set your Wincon's IP address & its Subnet Mask.

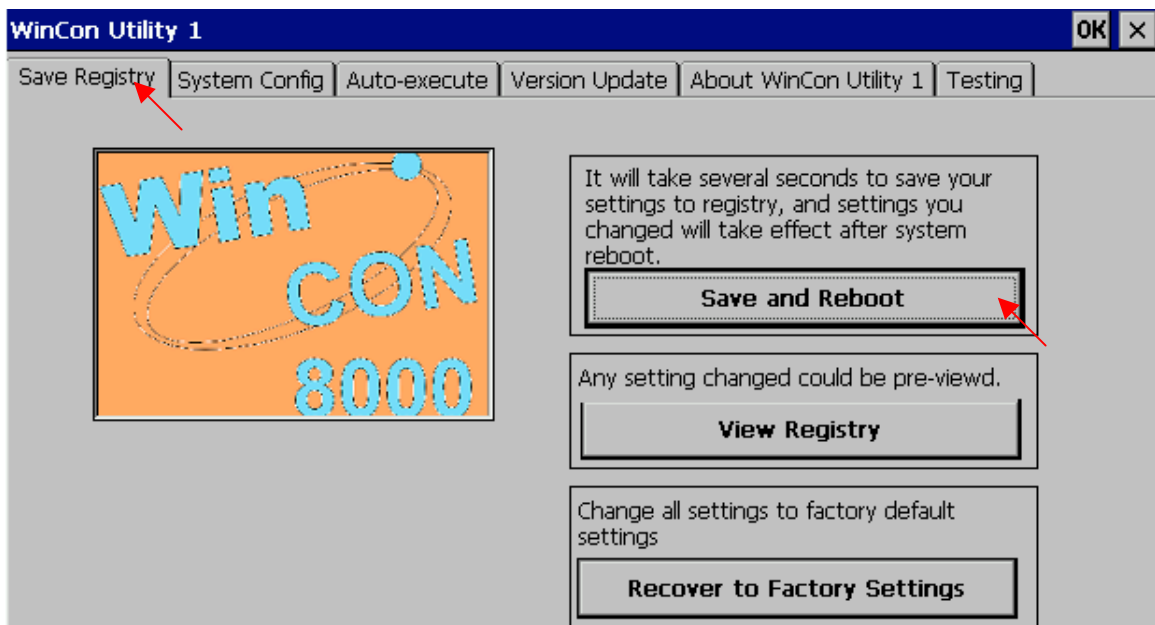


- C. Please run "Start" – "Programs" – "Wincon Utility"





Save Registry. This step will re-boot your Wincon controller.



3. Download **isawincon.exe** , **rs_whmi.exe** , **mscorlib.dll** , **QuickerNet.dll** , **Quicker.dll** (& **license.bin** if your Wincon is W-8031/8331/8731 or W-8039/8339/8739) from your PC to your Wincon controller.

Then please copy them to “\Compact Flash\ISaGRAF\”. & then re-cycle your wincon’s power.

You may use below methods to download these files.

A. Using ftp:

Wincon:

Please run “Start” – “Run”, then type in “**services load ftpd**”

PC:

Please open browser and then type in <ftp://<IP address>>,
for ex. <ftp://192.168.2.17>
copy all of them & past it.



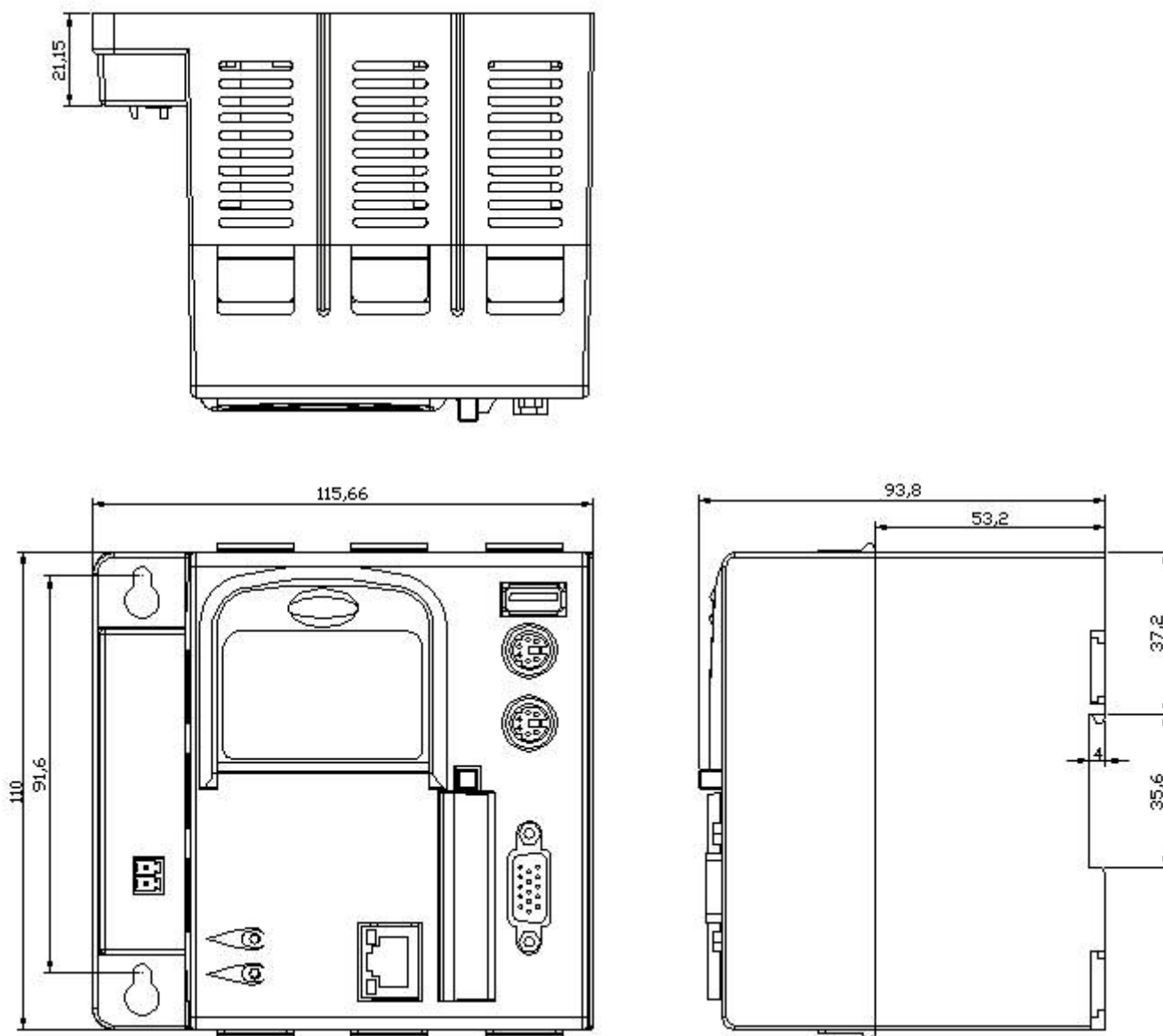
B. Using Pen Drive

Please copy **isawincon.exe** , **rs_whmi.exe** , **mscorlib.dll** , **QuickerNet.dll** , **Quicker.dll** (& **license.bin**) to “\Compact Flash\ISaGRAF\”

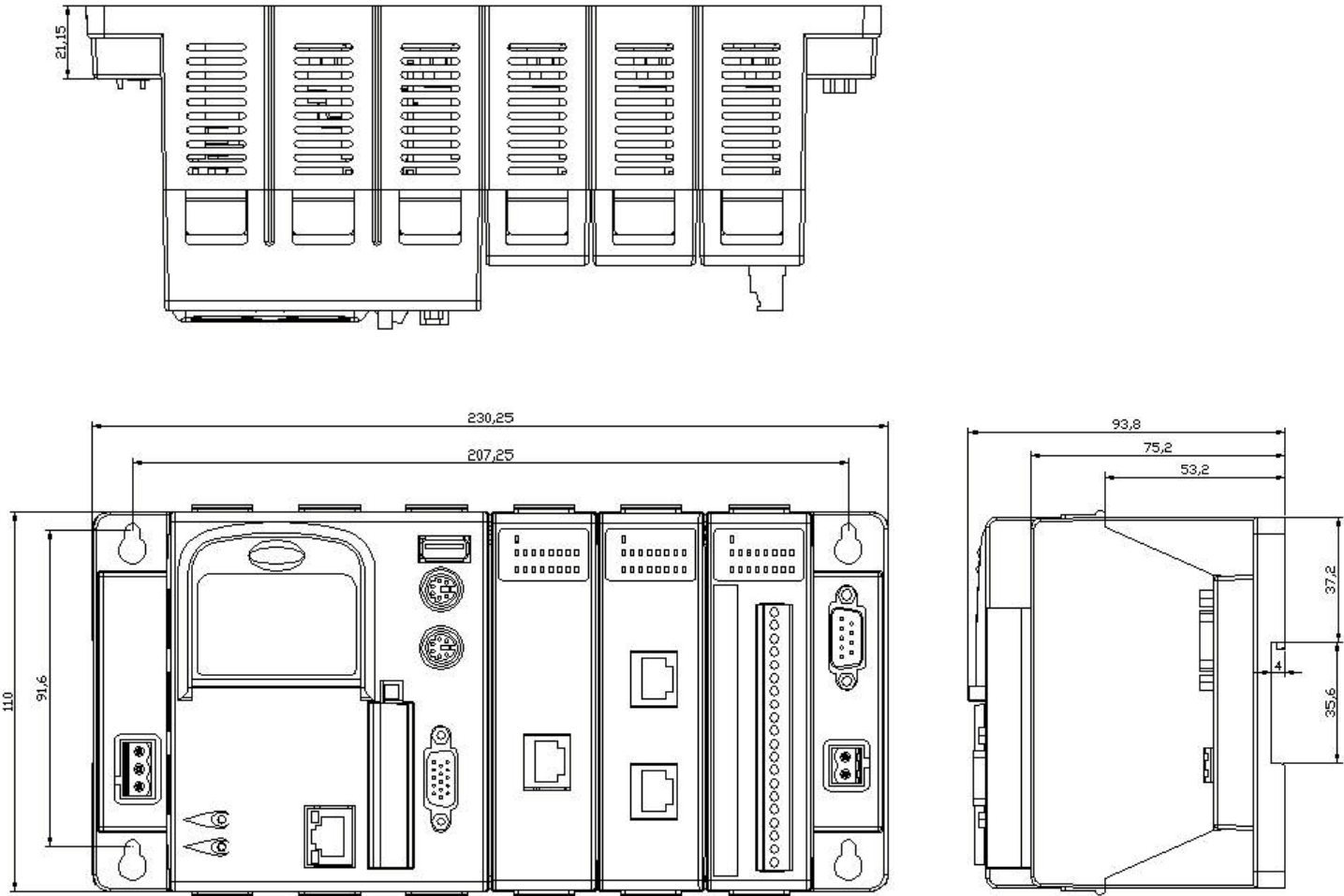
Remember to re-cycle your Wincon’s power.

Appendix C: Dimension

C.1: Wincon-8037



C.2: Wincon-8337



C.3: Wincon-8737

